UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

PROJECT OF LOW-COST MICROSCOPE AUTOMATION FOR DATA GATHERING

REPORT

Degree: Bachelor's degree in Aerospace Technologies Engineering
Delivery Date: 10/06/2019
Student: Ariadna Fernández Martínez
Project Director: David González Diez

Abstract

The AiScope is a project of a low-cost microscope that is used to diagnose global diseases in isolated communities. The original microscope is designed to be like any other optical microscope but is built in accessible materials to allow the user to build and use it in any place.

This project will consist on the development of the next iteration of the AiScope, which will incorporate an automatic system of sample analysis and image gathering that contributes to an easier diagnose of diseases. This system will allow any person with no technical training to use the microscope and obtain the sample images.

The automation process will consist in different parts. The first part will be a software (that will be released open-source in the future) that will determine if the sample is focused or not. This software is based in the border detection image processing technique, that will be explained later in this document and that allows to determine with precision if the sample is focused or not. The second part, the hardware, will consist in a series of moving mechanisms to displace the sample inside the microscope to focus it and obtain images of the whole sample. This hardware will be guided using a Raspberry Pi that will contain the created software. Finally, an Android App to use the system with a mobile phone will be developed. This App will be able to capture the images of the sample and to share information with the microscope.

In the end of the project, the user should be allowed to control the microscope with their mobile phone while the microscope focuses and recompiles the images of the sample without any human intervention. However, this will not be the final iteration and many improvements will be needed before its performance is optimal.

Resumen

El proyecto AiScope consiste en un microscopio óptico de bajo coste diseñado para diagnosticar enfermedades infecciosas en zonas aisladas. Actualmente, el microscopio está diseñado como un microscopio convencional fabricado con materiales accesibles para que cualquier persona pueda construirlo y usarlo.

Este proyecto consiste en el desarrollo de una serie de mejoras implementadas sobre este microscopio para permitir el análisis automático de las muestras y la creación de un banco de imágenes que contribuya a una más fácil detección de las enfermedades. Mediante este sistema, cualquier persona sin formación técnica podrá usar el microscopio para obtener imágenes de las muestras.

El proceso de automatización del microscopio está formado por distintas partes. En primer lugar, se crea un software (que está previsto distribuir de forma libre en el futuro) para determinar si la muestra está o no enfocada. Este software se basa en el método de detección de bordes para procesado de imágenes, explicado en la memoria del proyecto y que permite determinar con exactitud el enfoque de la muestra. Para implementar este software, también se diseña un hardware que incluye todos los elementos necesarios para controlar el microscopio de forma automática mediante una placa Raspberry Pi. Este sistema incluye una serie de servomotores que desplazan la muestra colocada en el microscopio siguiendo las órdenes del programa. Finalmente, se crea una aplicación para teléfono móvil que permite tomar las fotos de la muestra y compartir información con el microscopio.

Con este desarrollo, al final del proyecto, el usuario debería poder controlar el microscopio a través de su teléfono móvil y que este enfoque las muestras y recopile las imágenes de forma autónoma. Sin embargo, cabe añadir que esta no será la última iteración en el proceso de diseño del AiScope y que en las próximas aún habrá que introducir mejoras tanto de diseño como en el proceso.

Contents

List of figures

List of tables

Chapter 1

Introduction

Object

The aim of this project is to automatize the moving parts of the AiScope microscope in order to make possible an autofocus and to analyse the samples automatically without moving them in order to create an image bank. That would make possible to untrained people to use it and simplify the diagnosis of global diseases by having a wide collection of sample images taken in different conditions.

Scope

The scope of this project will be focused in the automation of the microscope platen whilst it is adapted to the new design of the microscope parts. The main points that must be accomplished during the realisation of the project are the following ones:

1. Install a servomotor to autofocus the objective.
2. Install a convenient number of servomotors to allow the microscope to analyse the sample thoroughly without the need of human intervention.
3. Make the system, first, able to work with a camera that would take pictures of the sample automatically, but also to make the camera removable so, in the end, it can be replaced by a mobile phone that would take the pictures with its camera.

With these simple points, one can have a clear idea of which parts will have the project. However, it must be mentioned that this project is included in a major project that involves a larger number of people. This project will be performed over an existing prototype, that can be seen in Figure 3 and will be done in collaboration with other project members that are working in other parts of the AiScope microscope, especially with Joana Aina Martorell, who will be redesigning the microscope [1].

Requirements

The main requirements for this project are the following ones:

a. Keep the changes and additions cheap so as the microscope still can be affordable to be used in isolated places.
b. Use pieces and materials that can be found in any place in the world in order to make the microscope able to be repaired in any place where it needs to be used.
c. If any pieces were designed, make them able to be cut with a laser or design them to be printed in 3D. However, pieces have to be thought to be able to be handmade cut
d. Add parts to the microscope without making it less movable or less practice to use. The microscope must be able to be transported without enough care and not to break.

e. Make the microscope easier to use; the objective is to make it automatic so untrained people would be able to use it properly.
f. More concretely, concerning to the automation:
    i. Make the microscope able to focus the images itself.
    ii. Make the microscope able to move the sample horizontally so it can be taken photographs of the whole sample without the requirement of a person moving it. Thus, the microscope must be able to distinguish different areas in the sample and take a picture of every part.
    iii. Adapt the microscope visor so it can be used with a camera but that camera replaceable because the final objective is that it can be used with a mobile phone camera.

Justification

The realisation of this project, the AiScope, appears from the need of underdeveloped countries of better diagnosis techniques to eradicate global diseases, more concretely and specifically: malaria; that currently can be cured but that needs to be early detected and treated [2].

Some of these global diseases, among them malaria, require microscopic techniques to be detected and there is where the need of this project comes up. Nowadays, laboratories have advanced technologies more than good enough to detect those diseases, but the problem resides in the environment where they are used. Laboratory microscopes work perfectly in a clean and controlled environment where the samples are previously prepared before being analysed and trained people are treating them. However, in isolated communities neither the environment nor the working conditions are the proper ones to use and preserve microscopes, and when they or one of their parts break, they become useless. Moreover, when the samples are not correctly prepared, due to a lack of materials or an incorrect preparation the parasites cannot be correctly detected although the means are available.

Due to these unfavourable environments, the evolution on the eradication of malaria in the past years has been slow and it still affects several countries. This evolution can be seen in the following map, published by the World Health Organisation in 2017 [3]:

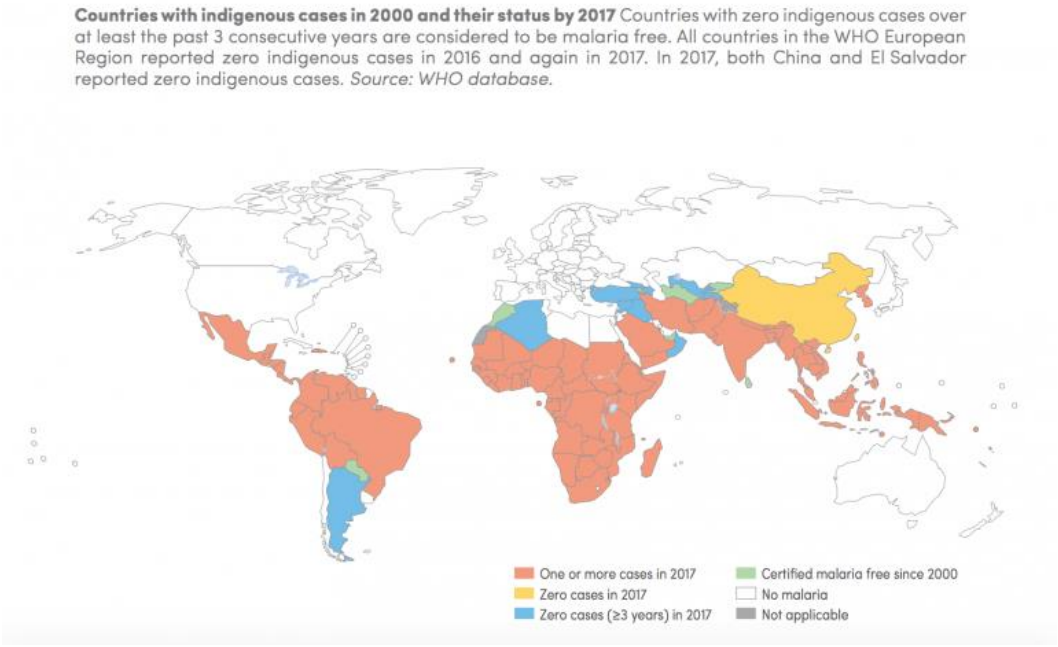


Figure 1 - Malaria extension in 2017. Source: WHO

It can be seen that, even though there are many countries that could be declared malaria free in 2017, still there are several countries in Africa, Asia and America where people get malaria and can die from it.

For this reason, this project pursues the objective of creating a microscope that, even though it will not have a high-resolution technology as professional optical microscopes have, will be enough precise to detect infectious diseases. In addition, this microscope will be cheap, so more instruments can be sent to isolated places and in case of break it can be replaced with a low cost; it will be portable and will not need trained people to use it.

Many prototypes have already been tested in the places and have given good results even though they still require manual adjustment and there is not enough data to detect those diseases. Thus, to improve the actual microscope this project, which will be a part of the global AiScope project, will be realised. As it was mentioned in the aim of the project, it will consist in the automation of the microscope, so it does not need of trained people in order to analyse the samples. The whole automation will have to be done following the requirements, which are the critical elements of the project, and that are compulsory to be satisfied in order to the project to be successful.

Chapter 2

Development

State of the art

Referring to the aim of the project, the final objective of the microscope that is going to be upgraded is to help in the diagnosis of global diseases, specifically, in the diagnosis of malaria.

The most important factor in the detection and treatment of malaria is that the diagnose can be made early and accurate [4]. Currently, there are different high-quality diagnosis methods that are useful to the correct treatment but as this project will be about microscopy; the state of the art will be focused in microscopy methods. Even though, must be mentioned that other methods such as rapid diagnostic tests (simpler) and molecular detection that are effective and are being used nowadays.

Microscopy diagnosis for global diseases

There are different microscopy techniques to diagnose malaria and its different parasites [5] but the most common involves Giemsa as a stain and light microscopy.

Giemsa Stain is a hematologic stain which is commonly used to stain hematologic samples [6]. It allows differentiating different malaria species and it combines Mytilene blue, azure A and azure B to colour cells. The most critic characteristic of this stain is its pH, which compulsory needs to be between 6.4 and 6.9. Furthermore, to obtain the best results, the formula needs to be fixed with a buffer solution, commonly formaldehyde. In the following images, the preparation of the samples with Giemsa Stain can be seen:

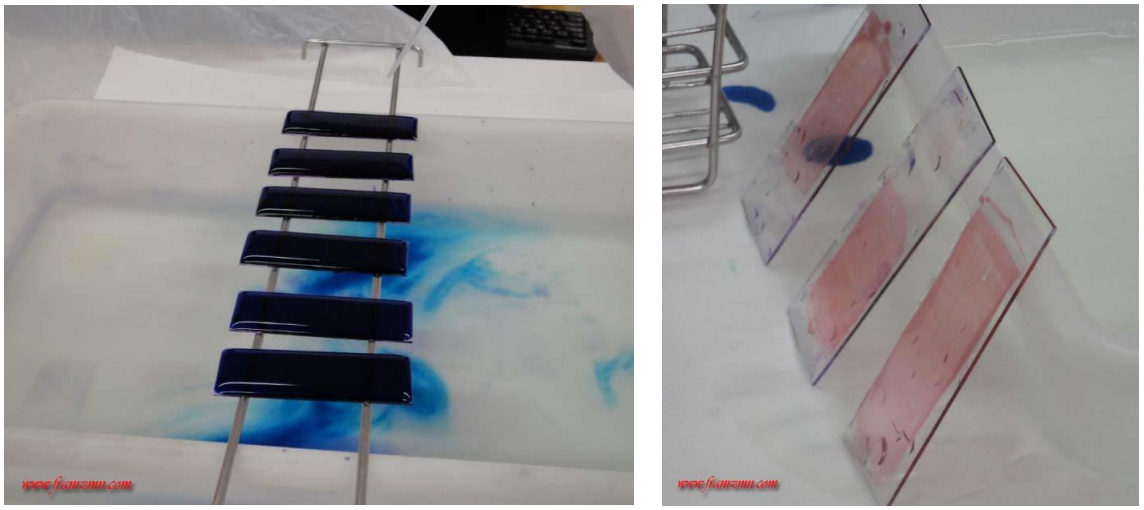


Figure 2 - Preparation of blood samples using Giemsa Stain. On the left, the samples with the stain; on the right, the samples once they have been stained. Source: Laboratorío clínico y biomédico [7].

Due to these conditions, it is normal that in an environment that is not controlled, the samples turn out to be useless because the pH of the stain is not adequate, or the washing of the sample is not properly made. This must be made watched carefully because bad preparation of the samples can induce to a false positive diagnosis [8].

In case Giemsa Stain is not available, Wright's Stain can be used instead but with this stain the identification of the different species can be difficult [9] However, it is useful because it contains methanol that makes that it does not need to be fixed..

The use of stains is actually a problem that is currently faced in microscopy diagnosis in isolated places. The main problem is that it is usual that stains are not properly preserved or used and the visible results are not the expected ones.

On the other hand, microscopes of x100 or more augments are compulsory to detect malaria parasites in blood samples. These microscopes are normally found in laboratories but in isolated places, they cannot be used in the proper conditions and they also cannot be repaired, which is a huge problem whenever they or any of their parts break. In addition, these microscopes require of high skilled specialists to perform the diagnosis. However, in countries like Papua New Guinea this is currently trying to be solved by training locals into malaria testing and diagnosis [10]. This has helped to double the number of certified microscope technicians and guarantee the diagnosis of at least 80% of malaria cases.

With this microscopy method, in laboratory conditions, a threshold of 50-100 parasites/mL can be found. If this value is estimated for the observations in worse conditions, it drops to 4-20 parasites/mL, which can be even more reduced if conditions of poor skilled technicians, bad preparation of the sample, bad quality of laboratory supplies or bad condition of the microscope are given; but even though, it could be found a detectable amount of parasites [8].

Nevertheless, using the microscopes out of the laboratory environments can lead to the damage of the systems due to the conditions and usage that it has to bear. The main problem in this case is that when any part is damaged is really difficult to substitute it and microscopes lose their functionality.

Given all these conditions, this section can be concluded by saying that thanks to diagnosis methods, malaria's incidence has dropped by 21% in the last years. Between the different diagnosis methods, it must be highlighted that microscope diagnosis has many advantages, such as being a technique known by most laboratories and hematologic tests being easy and routine. In addition, microscope tests can be done at any time and detect the disease although the patient had not presented symptoms [11]. However, the main disadvantage of using microscopy for diagnosis is that results are only 100% reliable if performed in laboratories and this is a huge problem when it comes to diagnosis in isolated zones [9].

Automatic microscopy

The automatic acquisition of images for optical microscopies was developed several years ago and it already consists in systems that have a focus searching algorithm and a control to allow the movement of the samples in the three axes [12].

The initial proposed automatic microscope was based in a control system with speed reducing pulleys attached to the screws that regulated the position in a conventional microscope. This system was controlled with a computer, which controlled at the same time the Web camera that was used to control the image of the microscope.

Then, apart from the mechanic system, an algorithm of focusing images would guide the microscope.

This algorithm was based in the border detection method of image analysis. The border detection method consists on analysing the image in areas and extract its characteristics. Its principal purpose is to identify the points of a digital image where the brightness changes drastically, or in a more technical way, it has discontinuities. The results of applying this method can lead to the detection of object borders, borders in the inner elements of the objects and inner discontinuities. With this data, it is possible to treat the most relevant information of the image conserving its structural properties.

Therefore, based on this method, different algorithms to measure the focus exist. These algorithms present a similar quality in the image analysis although they are the ones that require a higher computational time compared to others based on other image analysis methods.

Currently, there are several parts of a microscope that can be automated and not only the focusing process [13]. Apart from that, the microscope can incorporate shutters that block the light sources from illuminating the specimen between camera exposures, wavelength selectors that apply different filters to the images or different illuminating sources that can include lasers or arc gas lamps for fluorescent excitation.

These automatic systems are extremely valuable in biologic analysis because they are really useful when many repeated observations are required for both live cell imaging and high throughput analysis. Thanks to this, a larger number of samples can be analysed in a single day turning the classic time-tested process into an overpowering force that can turn the analysis in a faster and more efficient technique.

Initial approximation

The first model of the microscope is a basic model, which does not contain any electronic parts. It is based in three rudimentary manual mechanisms to make it able to move the sample in the three axes.

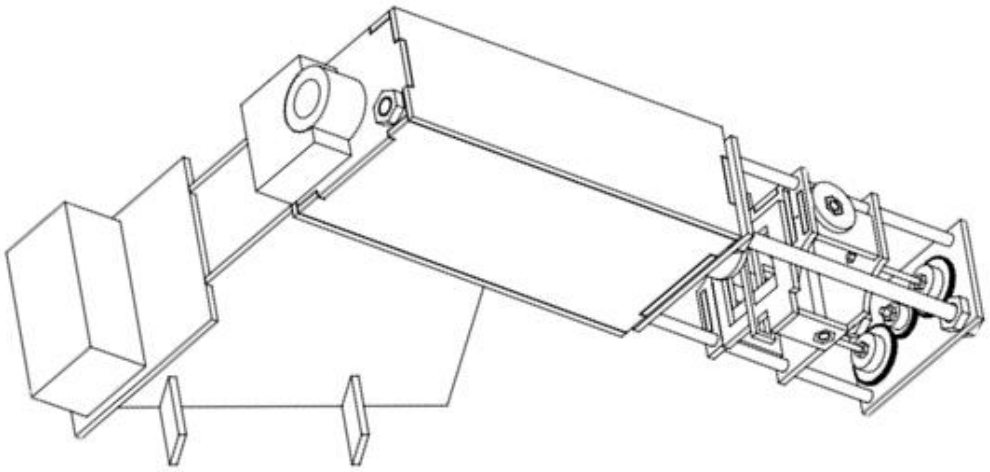


Figure 3 - Scheme of the initial microscope. Author: Joana Aina Martorell [1]

In this scheme, the three mechanisms can be evaluated. In first place, the two inferior threaded rods make it able to move closer and further from the lens. This system incorporates a series of gears that allow a reduction in the displacement.

The user must turn the white wheel that is attached to the smaller gear, whose movement is transmitted to the bigger ones, at the same time. Those bigger gears make the longer threaded rods to turn and move the part that contains the sample.

The other two mechanisms are attached to the part that contains the sample. Both are very similar, and their function is to move the sample horizontally over a plane. They consist in a white wheel attached to a threaded rod, which is fixed, that moves one of the pieces that subjects the sample by screwing on that piece.

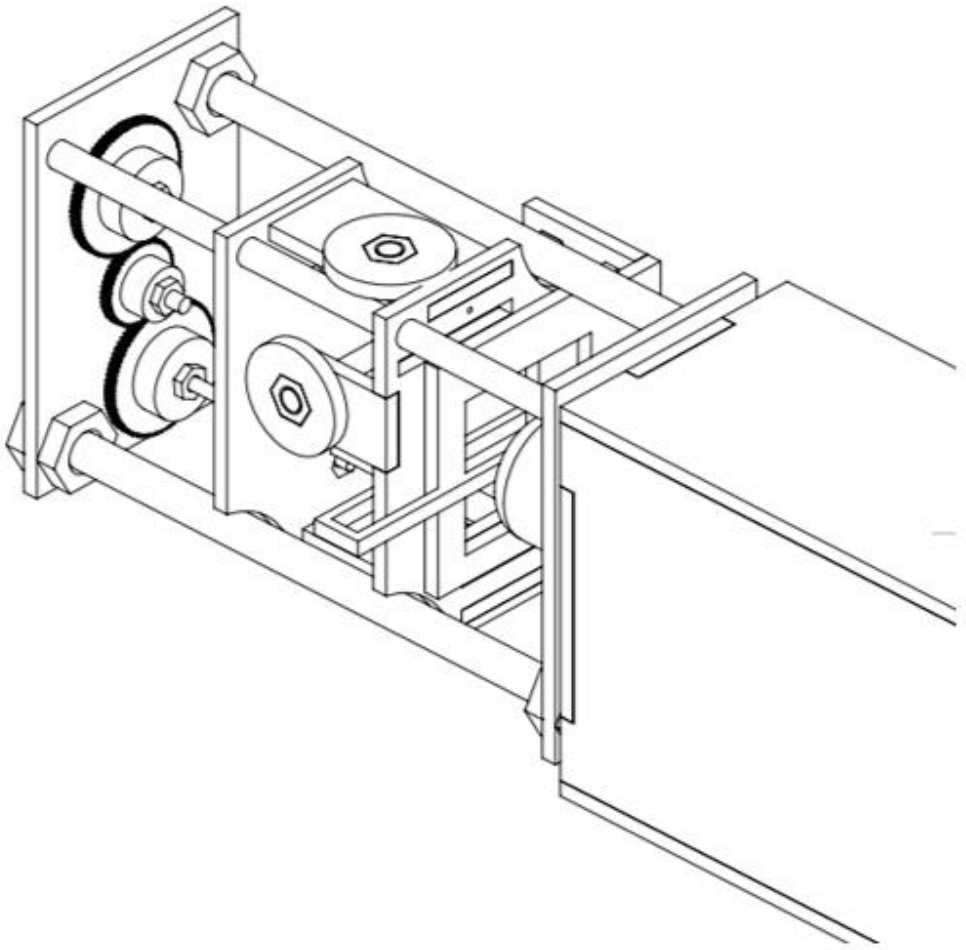


Figure 4 - Scheme of the mechanisms in the original microscope. Author: Joana Aina Martorell

Given that the wheels are accessible and easy to replace, it does not seem a problem having to replace them with the servomotors, but it must be remembered that the final microscope must be compact and not have easy breakable parts. Therefore, the current structure can be conserved, but the servomotors must be incorporated organically so they do not disturb to the correct performance of the microscope.

Selection of alternatives

This project is based on the model that was built from the previous iteration of the microscope. Therefore, there are not many things that can be designed or remodelled. Furthermore, considering that another student is also working on the redesign of the prototype [1], changes in the structure are conditioned to many factors.

Thus, the principal changes that will be added in this case, are the already commented ones, which will consist on the addition of a series of motors that will move the sample in the three axes and the corresponding electronics.

In order to allow this motion, some elements will be necessary, and these are the ones that will be presented in this section. Notice that not only the used items will be presented but also the alternatives that are considered and even tested.

Motors

The first option that was considered were three output standard servomotors, which are shown on the left in the image below. These motors usually allow only a closed degree range; they are thought to control parts that are static and only need to turn in one axis a certain number of degrees.

Initially this is not useful for the case that is contemplated, at least for the main part that consists on focusing the sample, in here but as they can be modified to allow a complete turning and they provide a reduced velocity; they are considered as an option.

They require a PWM[1] channel, which is also beneficial for their working, as this method allows a precise control of the voltage that is given to the output. At the same time, those servomotors only need of a single output connection and the voltage connection, which allows a reduction of the electronic items that, must be added into the microscope.

Continuing with the servomotors, an option that has to be considered is using continued turn servomotors that allow a 360º turning and more than a single turning. Those servomotors will be evaluated because there are two ways of obtaining a continued turn servomotor. The first way, that would be easier but more expensive (the difference in prices has to be taken into account because it may be more useful to increase the budget than add more complexity to the building process of the microscope) would consist in simply buying the servo. The other way, which is considerably more complicated and homemade (which can conduce to imprecisions), consists of modifying a standard servomotor that allows a 180º turn. This process will not be explained as a part of this project as it is already explained in some blogs [14] and has not been developed for this project realisation. However, this is also a good option to consider if the future manufacturer could only find 180º servomotors, they would be able to modify them and still build the AiScope.

Then, the second option was to try standard small electric motors, which are shown on the right in the image below, and that can work with both PWM and GPIO[2]. These motors are simpler than the previous ones and have the main advantage that can rotate in both directions easier than the servomotors as the outputs are specified. These motors are always controlled via a microprocessor that will be presented later, due to the lack of protection of the raspberry pi outputs (this means

[1] The Pulse-Width modulation method consists on reducing the average power given by an electric signal by breaking it into discreet parts. It is usually employed to control analogic circuits with microprocessors' digital outputs. For the motors, it is useful to control their speed.
[2] General Purpose Input/output

that any peak of voltage or amperage produced by the motors' external batteries could burn the Raspberry Pi).

However, these motors present the following main problems:

a) Their speed is much faster than is needed. Their speed can be controlled reducing the voltage, but reducing the speed, the torque is also reduced and, consequently, the force that it can make on the first axis and has, later, to be transmitted via gears to the next axes. Therefore, the only solution is to drastically reduce the time the motors are turning but this can lead to imprecisions.

b) These motors require of two GPIO outputs, which would not be a problem in this case. However, if one wanted to use the Raspberry Pi for other uses or to add additional features to the AiScope that required of GPIO pins, it could condition the connection of all the items that were required. Furthermore, the connection of these three motors with the microprocessors implies a huge amount of cable that make the microscope more fragile and less ergonomic and portable and also adds the need of incorporating a battery, which implies extra items to carry and extra weight.
This also means that one part of the project will consist in the addition of all the systems required in a way that cannot affect to the correct operation of the microscope.

c) Their weight is also considerably higher that the servomotors' and can present a problem in the joining of the motors to the microscope, especially if their position is not favourable.

As it has been commented, in the following images are presented the two main alternatives considered, that will be tested during the implementation of the project.

Figure 5 - Servomotor

Figure 6 - Miniature motor

Raspberry Pi

The Raspberry Pi has different models available that present different performance possibilities. First, a table to summarize the characteristics of each model will be presented [15]:

Table 1 - Comparison of Raspberry Pi models. Source: Luis Llamas

Model	Raspberry Pi 1 B+	Raspberry Pi 2B	Raspberry Pi 3 B	Raspberry Pi Zero	Raspberry Pi Zero W
Prize	30	35	35	5	10
SOC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2835	Broadcom BCM2835
Core	ARM1176JZF-S	Cortex-A7	Cortex-A53 64-bit	ARM1176JZF-S	ARM1176JZF-S
Nº of cores	1	4	4	1	1
GPU	VideoCore IV				
CPU Clock	700 MHz	900 MHz	1.2 GHz	1 GHz	1 GHz
RAM	512 MB	1 GB	1 GB	512 MB	512 MB
Memory	Micro SD	Micro SD	Micro SD	Micro SD	Micro SD
USB	2	4	4	1 MicroUSB	1 MicroUSB
Ethernet	Yes	Yes	Yes	No	No
Wi-Fi	No	No	Yes	No	Yes
Bluetooth	No	No	Yes	No	Yes
HDMI	Yes	Yes	Yes	Mini	Mini
GPIO	8	17	17	17	17
UART	Yes	Yes	Yes	Yes	Yes
SPI	Yes	Yes	Yes	Yes	Yes
I2C	Yes	Yes	Yes	No	No
DSI (LCD)	Yes	Yes	Yes	No	No
Camera	Yes	Yes	Yes	Yes	Yes
Height	85.6mm	85.6mm	85.6mm	65mm	65mm
Length	53.98mm	56.5mm	56.5mm	30mm	30mm
Width	17mm	17mm	17mm	5mm	5mm
Weight	45g	45g	45g	9g	9g
Amperage consumption	700mA	820mA	1400mA	350mA	350mA

In this project, the Raspberry Pi 3 and the Raspberry Pi Zero (and Zero W) will be considered. The other models will not be discussed as their characteristics do not make a real difference that make it worth it to test them.

The Raspberry 3 [16] is considerably faster and more capable than Raspberry Pi Zero [17], which is the most basic computer of the whole gamma, also the smallest one.

For this project, which needs of considerably small computer power and is not complex in a level of coding, the Raspberry Pi Zero should be enough to make the microscope work. Furthermore, this model is considerably cheaper than the other ones (which is a requirement of the project). The Raspberry Pi Zero and Zero W have an approximate cost of 5-10€ (30€ if it comes with the whole pack of connectors), whereas, the Raspberry 3 costs 35€ (85€ if it comes with the pack).

Given that considerable difference of prices, the Raspberry Pi Zero W is expected to be chosen for the final microscope design. It will be chosen the model Zero W instead of the Zero because it does have Wi-Fi and Bluetooth, which will probably be necessary. However, at least, while the project is in development, a Raspberry Pi 3B will be used because this one is faster, more powerful and has a normal HDMI connector, which makes it easier to have it connected to a screen.

Apart from the Raspberry Pi gamma, Arduino was also considered as an option, but was soon discarded because an Arduino board needs of a computer to work while the Raspberry Pi is a computer itself. Needing to carry a computer to use the Arduino would make the microscope less portable and obviously more expensive to use. Nevertheless, it must be remembered, that even though the Raspberry Pi is a computer itself, it does need to use electricity to be operative, and this represents a handicap to the automation of the microscope. This is not a competence of this project, but, in the future, if this automation is implemented, another way to power the electronic system could be studied or if not, a way to make the system self-sufficient.

In the following images can be seen the Raspberry Pi 3 and the Raspberry Pi Zero. Notice that the difference between their sizes can present also an advantage when it comes to placing the board inside the microscope.

Figure 7 - Raspberry Pi 3 B

Figure 8 - Raspberry Pi Zero

Camera

This section will not be exactly a selection of alternatives but a series of alternatives that will be used throughout the project to develop it.

The camera has a main role in the focusing process, so to start and to make it simpler, the first iteration of the system is built with the Raspberry Pi Camera Module [18].

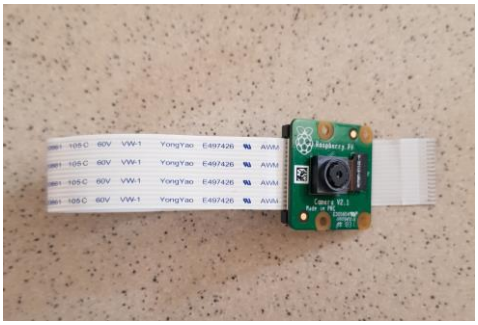


Figure 9 - Raspberry Pi Camera Module V2

This camera allows the Raspberry Pi to take the images itself and does not need any additional software to work. This camera has enough quality for the system to work properly.

When the software and the system are properly working and been tested, this camera is replaced by a mobile phone that will at the same time connect with the Raspberry Pi. This is what will make the microscope portable and easy to use. It is thought to be used with any model of phone as the camera and then, the images taken be stored on the phone too so, later, with an app that will be developed, it can create a catalogue with all the taken images and detect the diseases automatically.

Working system

The central point of the project is the Raspberry Pi computer. Initially, the Raspberry Pi is controlled as a computer, with a screen and a keyboard, but later, it is adapted to run the AiScope program when it is initialised. In its final configuration, the Raspberry Pi is expected to need no control at all.

The Raspberry Pi 3B has the following pin configuration [19], along which the connections will be distributed.

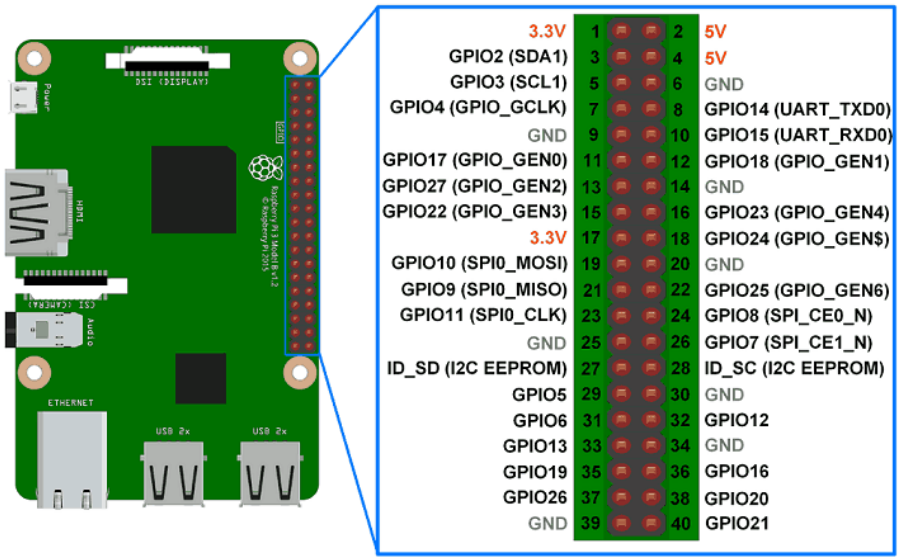


Figure 10 - Raspberry Pi 3B pin distribution. Source: ElectronicWings

This configuration can also be controlled in the command window of the Raspberry Pi with the following command:

```
pinout
```

Which provides the following information:

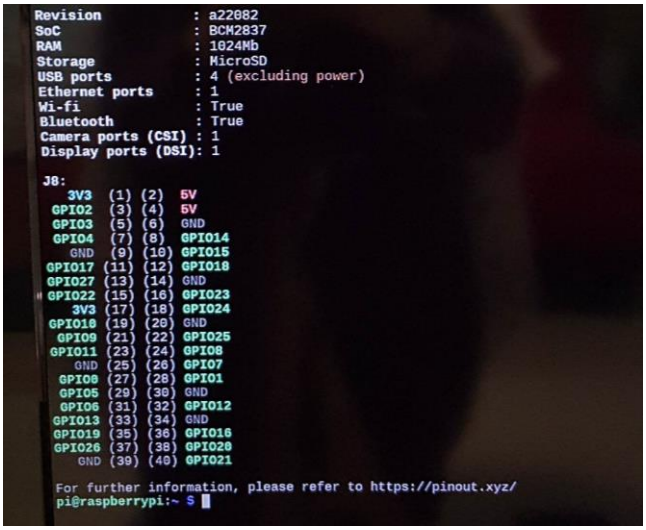


Figure 11 - Information obtained with the command pinout

This information is useful because it verifies the position of the pins and avoids confusion between Raspberry Pi models.

Changing to the Raspberry Pi Zero, or Zero W, the pin distribution is exactly the same.

Therefore, from the Raspberry Pi, there will be five principal outputs: one will be the led that enlightens the sample; the second one is the phone that will be connected via USB/internet to the Raspberry Pi so the information can be shared. It must be considered that this phone will not be present in the preliminary phases of the project because the Raspberry Pi camera will be used to make the process of taking pictures easier. However, those ones will be present in any case. Finally, the missing three outputs are for the motors. The motors will be chosen after they are tested and, at is has been mentioned before, there are two options. The first one involves the microcontrollers that, at the same time, are controlling the motors. There is a huge range of microcontrollers but in this case, it will only be tested one model because they are used only for testing and it is not sure that they will be used. If they were chosen as the best alternative, in future iterations it could be studied the convenience of a concrete model of microcontroller. The second option involves servomotors, which do not need of microcontrollers.

Once the outputs are presented, their respective connections will be explained.

Led lights

The connection of the led has no complication. Leds usually work with a voltage of 2.1V so, as the Raspberry Pi provides a voltage of 3.3V in their lower voltage sources, a resistance is needed.

If it is considered that the voltage given by the Raspberry Pi board is of 3.3V, it is required that the resistance produces a voltage drop of 1.2V (the different between the provided and the needed). Also, the amperage needed is of 20mA, so the required resistance (obtained with Ohm's law) will be:

$$R = \frac{V}{I} = \frac{1.2V}{20mA} = 60\Omega$$

That will be rounded to 100Ω to make it easy to find a replacement of the resistance in case of any problem. Furthermore, the use of a slightly higher resistance will make that the light provided is not the most intense and that will make the sample to be seen more clearly.

Then, the connection is simple. The cathode of the led must be connected to a ground pin of the Raspberry Pi and the anode must be connected to a GPIO[3] pin with the resistance in between. In the following image, one can see those connections graphically represented [20]:

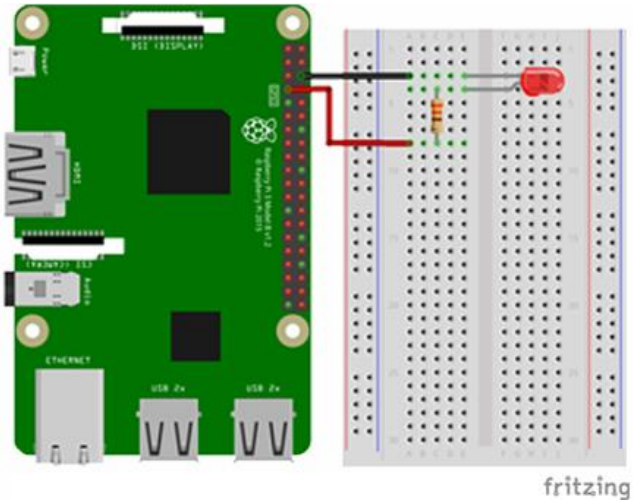


Figure 12 - Connection of the led to the Raspberry Pi. Source: Programo Ergo Sum

Motors via microcontrollers

Then, there are the microcontrollers. As it has been presented before, the motors need to be controlled through a microcontroller to protect the Raspberry Pi from undesired peaks of current. In this case, there are plenty microcontrollers that could be evaluated but only will be presented the one that was chosen for the tests.

In this project, the microcontroller that is going to be used for testing is the L293D. This microcontroller allows connecting to two motors and has a range of voltage of 3.3V or 5V depending on the enable voltage that is chosen. It also allows controlling the voltage that is applied to the motor depending on the battery that is connected. Moreover, the fact that the battery can be chosen from 0V to 3.3 or 5V allows that any kind of battery can be used to run the system and makes it universal.

[3] It is connected to a GPIO pin instead to a voltage output pin so it can be controlled when the light is on.

For the testing of the project, two standard batteries of 1.5V are used, but any battery in the voltage range can replace these batteries.

The L293D controller has the following structure [21]:

Figure 13 - Microcontroller L293D

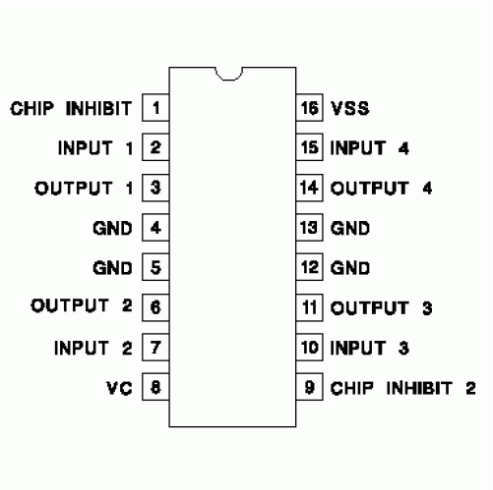


Figure 14 - Outputs of the microcontroller L293D. Source: Dinastía tecnológica

As it can be seen, the microcontroller needs to be connected to two voltage outputs of the Raspberry Pi, and to two ground pins. It also needs to be connected to the batteries. In relation to the batteries, it must be clarified that the two microcontrollers would make use of the same battery pack because in no moment the two microcontrollers would require the power at the same time.

The connection of the motors to the Raspberry through the microcontrollers can be seen in the following image [22]:

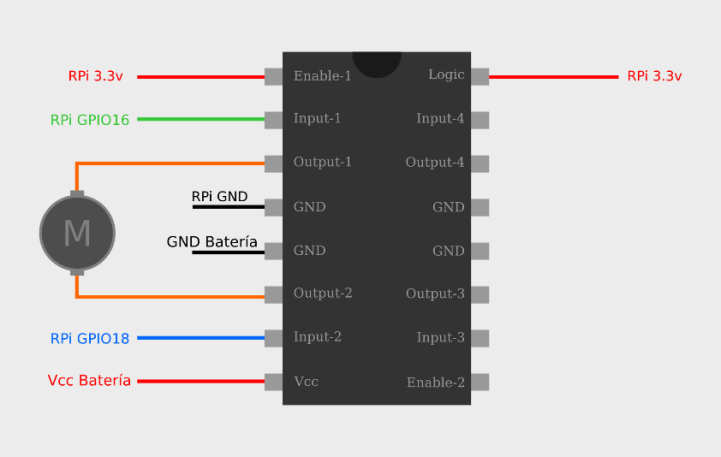


Figure 15 - Connection of the motor to the Raspberry Pi through the L293D microcontroller. Source: Roblogs

Following the last figure, all the connections can be made.

However, from here it can already be seen that this alternative will require of a considerably higher number of connections and items and; the most important, will suppose a high increase in the microscope total price (one can see the price of all the elements with further detail in the budget of the project).

Servomotors

Servomotors, unlike motors, do not require of a microcontroller to be connected to the Raspberry Pi. Servomotors only need of three connections: a voltage source, a GPIO pin of the Raspberry Pi and the ground.

Servomotors, as it has already been mentioned, are controlled through a PWM output that is why they only need one connection done to the raspberry pi. This single output provides of a signal that depending on its frequency and duty cycle generates a different kind of rotation (variating its speed or direction).

The connections that are required to control a servomotor with the Raspberry Pi can be seen in the following image [23]:

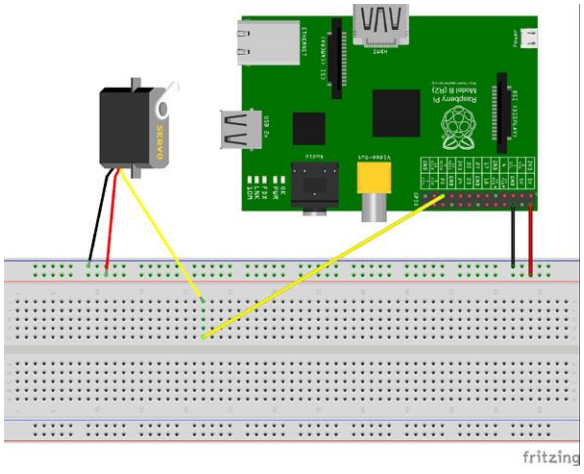


Figure 16 - Connection of a servomotor with the Raspberry Pi. Source: FPaez

In this case, the connections are much more simple than with the miniature motors and do not require additional elements.

Raspberry Pi Camera Module

The Raspberry Pi camera module only requires of the camera option in the Raspberry Pi computer to be activated so it can be connected. Its configuration is also simple because it just needs to be activated in the Raspberry Pi menu and then it can be accessed via Python IDLE or the commands window.

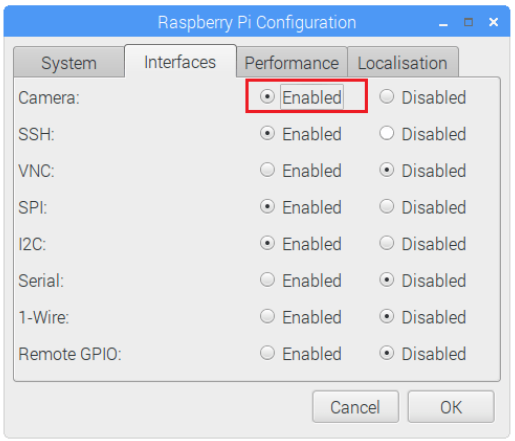


Figure 17 - Raspberry Pi configuration window

With that, the camera just needs to be plugged in in the camera input of the Raspberry Pi.

Mobile phone

Finally, the mobile phone is connected to the Raspberry Pi for both controlling it and take pictures instead of the camera.

The phone will be able to take pictures of the sample located in the following support, designed specifically for it:

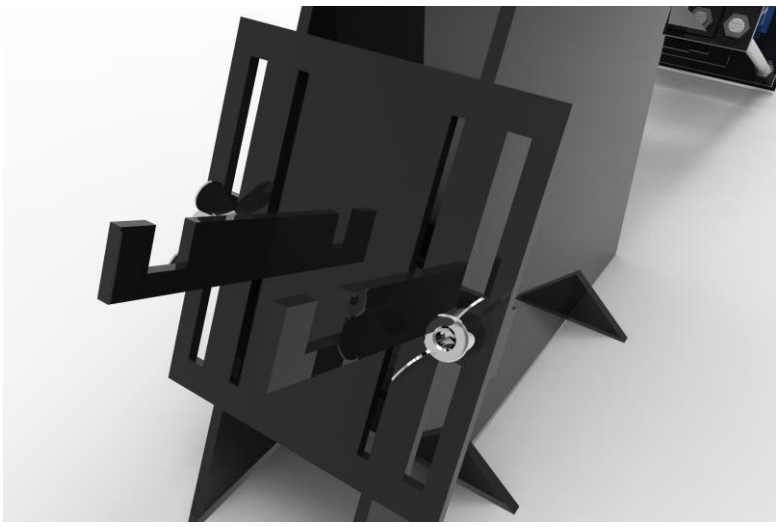


Figure 18 - Support for the mobile phone

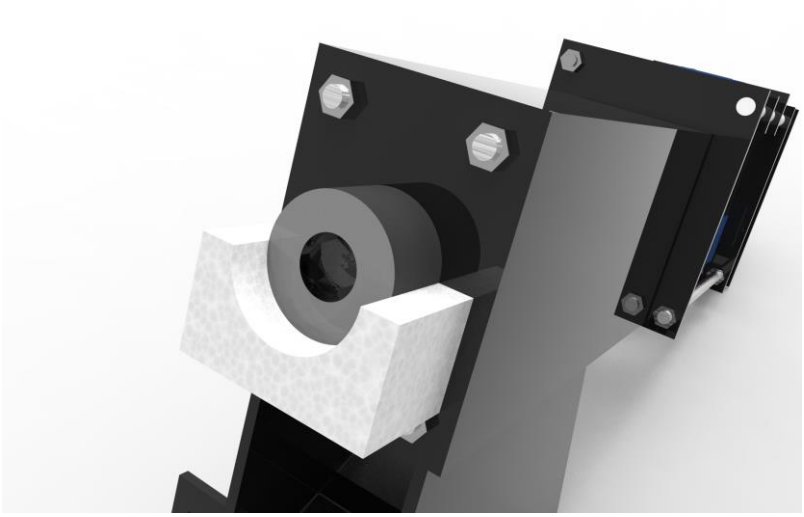


Figure 19 - Support in the ocular for the mobile phone

This connection will be treated later in the project as it is only incorporated in the final chapters as an improvement of the initial approach.

Different ways to connect the phone to the Raspberry Pi will be explored to determine which one can be the easiest one but at the same time the most appropriate for the usage of the microscope.

Chapter 3

Sample focus

The proposed solution of automation comes as a process, involving the hardware described in the previous chapter; that focuses the sample while it takes pictures of the entire sample.

The microscope is going to work then in two phases that will consist, on the one hand, on image analysis and, on the other hand, on the displacement of the sample that will be combined to create a main code. Thus, in order to make a clear explanation of the code and thez whole process that will be made, the theoretical basis behind the code will also be explained. In order to explain it, in this chapter it will be explained how it is detected whether the sample is focused or not and in the next chapter, will be presented the process of automation.

So, in the following points it will be explained how to detect if the image is out of focus.

Detecting focused images

The algorithm to detect if the images are focused is presented in the function Taking pictures of the sample block and analysing them. It works following the next process.

Adapting images for its analysis

The first step to focus the image is to know how it can be easily detected if the image is focused or not.

When the image is imported from the camera, the first thing that is compulsory to do is to turn it into the grey scale. This step is necessary because using images in RGB increases computational time exponentially. These images have three colour scales (red, green and blue), and working with them is like working with three images at the same time. With the images in grey scale, the colours are reduced to one scale.

This procedure consists in calculating an equivalent 'E' from the three planes of the image in RGB[4] [24].

$$E = \frac{R(x,y) + G(x,y) + B(x,y)}{3}$$

Therefore, what it does is to take the data from the imported image and process it to turn it into black and white.

However, this method presents a problem: images with a high content on red or green turn out darker than they were and images with high content of blue are lighter. Then, to solve this problem, many linear approximations are available. The linear approximation that is chosen depends completely on the usage that the black and white picture will have. In this case, the typical TV approximation will be used as this one is simple and vastly used.

[4] Abbreviation for red, green and blue

These values are the following ones:

$$W_R = 0.229$$

$$W_G = 0.587$$

$$W_B = 0.114$$

Therefore, the formula has the final form:

$$E = W_R * R(x,y) + W_G * G(x,y) + W_B * B(x,y)$$

However, this does not need to be implemented because in the python packages image and cv2 include options that already do it automatically. In the next function, adding the argument 0 when importing the picture transforms it into grey scale:

```
img=cv2.imread('image_path/image_name.jpg',0)
```

Obtaining the image gradient

Then, the next step is to find the image gradient. The image gradient is a generalization of the derivative for multivariable functions. Finding it will allow the program to find the sharpest zones in the image in both directions x and y.

In general, the gradient function represents the slope of the tangent to the function. More precisely, the gradient, points to the direction of the maximum increment of the function, its module is the magnitude of the slope in that direction. Because of that, must be considered that in images, the highest change from black to white will present a positive slope, and the change is from white to black, the slope will be negative.

Therefore, to calculate the value of the derivative, the following process must be done: the derivative is usually calculated from the Laplacian of the image or using first order derivative filters. In order to calculate the Laplacian, the first two derivatives have to be calculated. It is also possible to obtain the Sobel derivatives from the first derivative. Sobel derivatives give the value of the variation of the gradient in certain directions: a vertical direction and a horizontal direction. Both methods are valid and give accurate results, but no choice will be made before they are tested in the microscope.

Thus, on one hand, Sobel derivatives are calculated the following way [25] [26]:

a. Horizontal Sobel derivative (Sobel X): is obtained from the convolution of the image with a Kernel matrix of an odd size. Most common case is a 3x3 matrix.
b. Vertical Sobel derivative (Sobel Y): is obtained from the convolution of the image with a kernel matrix of an odd size, same as the previous one.
c. Convolution: is calculated with the following method:

   These are the Sobel matrix to obtain the gradient:

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Before continuing, one must find interesting to know where these matrices come from.

The process starts with the formula of the basic derivative of a function:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Even though, this one is based in the difference of the function forward. However, the central difference is used because it provides a higher precision.

$$f'(x) = \lim_{h \to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$$

Then, if a discreet horizontal signal is taken, the procedure is the following one:

Example signal:

$$[\,\mathbf{10}\ \ \mathbf{20}\ \ \mathbf{10}\ \ \mathbf{20}\ \ \mathbf{20}\ \ \mathbf{25}\ \ \mathbf{25}]$$

Then, the derivative for one point, for example, the fourth one would be:

$$f'(x) = \frac{f(x+1) - f(x-1)}{2} = \frac{20 - 10}{2} = 5$$

So, the following deduction can be made: the filter applied around the selected point is:

$$[\,\mathbf{-1}\ \mathbf{0}\ \mathbf{1}\,]$$

As it is possible to see that the term that is placed right before the point is summed in negative, the point itself does not appear and the term that is before is summed in positive. With this consideration, that provides a horizontal derivative, and a part of the filter that is used to make a weighted average and a scaling:

$$\begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}$$

So, if the derivative vector and the weight averaging and scaling vector are multiplied the Sobel derivative filter is obtained.

Once this is explained, the explanation of the process can proceed. Thus, the following step is to run the filter over the image:

$$\frac{\partial f}{\partial x} = S_x \otimes f \qquad\qquad \frac{\partial f}{\partial y} = S_y \otimes f$$

And the gradient has the following expression:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

That can be decomposed in the two following terms, the magnitude and the direction:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\theta = \arctan\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

On the other hand, the Laplacian filter is the following one:

$$G_l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

These Laplace filters start from the basic form of the second derivative:

$$f''(x) \approx \frac{\partial^2 f}{h^2} = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

The whole process to obtain the filter will not be repeated and the filter that is obtained with this expression is the following one:

$$[\mathbf{1} \quad \mathbf{-2} \quad \mathbf{1}]$$

It is operated as it was done with the Sobel filter in order to obtain the convolution matrix.

However, these filters are not going to be implemented as a part of this project and they will be applied using the available functions of the opencv python library. These are the functions that will be tested [27]:

Code of the functions:

```
#calculation of sobel x
Sobelx=cv2.Sobel(frame,cv2.CV_64F,1,0,ksize=5)

#calculation of sobel y
Sobely=cv2.Sobel(frame,cv2.CV_64F,0,1,ksize=5)

#calculation of laplacian
laplacian=cv2.Laplacian(frame,cv2.CV_64F)
```

These functions require of the following arguments:

a. Frame: the image that will be analysed
b. Cv2.cv_64F: the depth of the destination image. Using this one returns the same as the origin image.
c. (only for the Sobel) x and y orders: they set if the derivative is applied in the x direction or in the y direction. These values have to be compulsory one 0 and one 1.
d. Kernel[5] size: size of the kernel (can be -1, 1, 3, 5 or 7; the usual is 3 or 5).

As it has been explained, these functions will give as a result the images transformed into gradient matrices, which will have to be analysed in order to extract useful results.

Analysis of the gradients

The previous functions return as a result the images in a matrix form that still allows the representation of the image. As an example, here is presented an image taken with the Raspberry Pi camera in its original form and with the three filters applied:

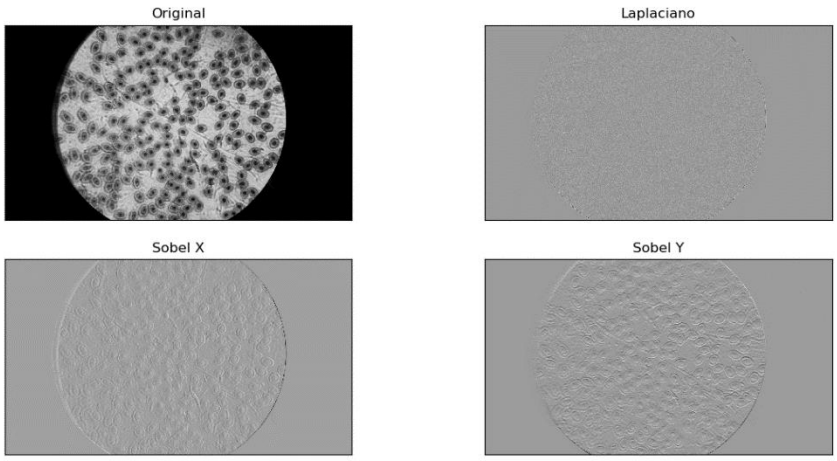


Figure 20 - Example of an image with the three filters applied

The program will initially be developed using both the results of the Laplacian function and the Sobel derivative functions, to be able to compare the results of the both functions. Nevertheless, the Laplacian function is believed to be more accurate than the Sobel filters because it works with zero-crossings instead of peaks. However, this also can present problems because the Laplacian results are also obtained with a higher noise. Thus, no method will be selected before a further analysis is performed.

Then, the first step was to obtain a criterion to detect whether the image that was focused or not. These criteria were stablished after the analysis of several images taken with the microscope. Some of these images will be presented below:

[5] In image processing, a kernel is a convolution matrix used for blurring, sharpening or border detection. The results are obtained by doing a convolution between the kernel and the image.

a) In this image, the camera captures a focused sample:

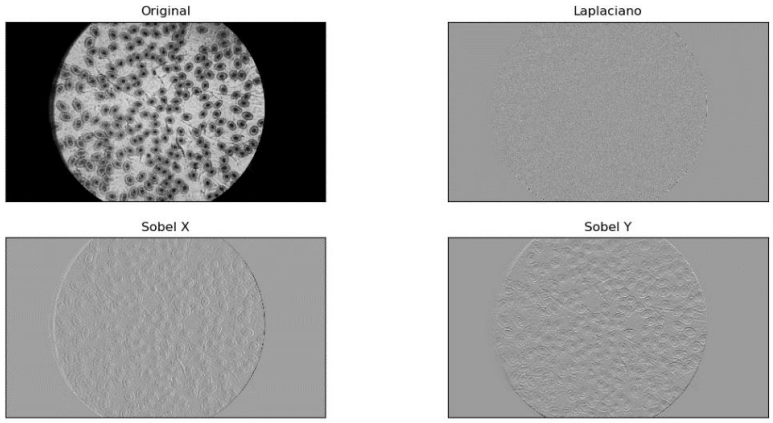


Figure 21 - Filters applied to an image of the sample correctly focused

b) This image shows an unfocused sample:

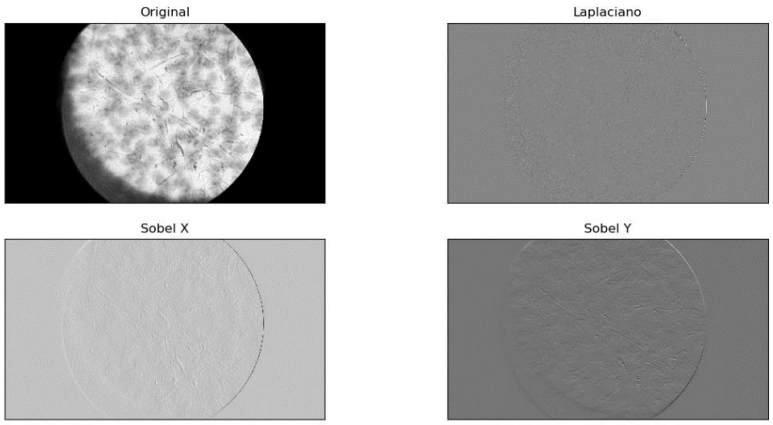


Figure 22 - Filters applied to an unfocused image

c) This image shows a completely out of focus sample (the sample was so far away from the lens that was not even captured):

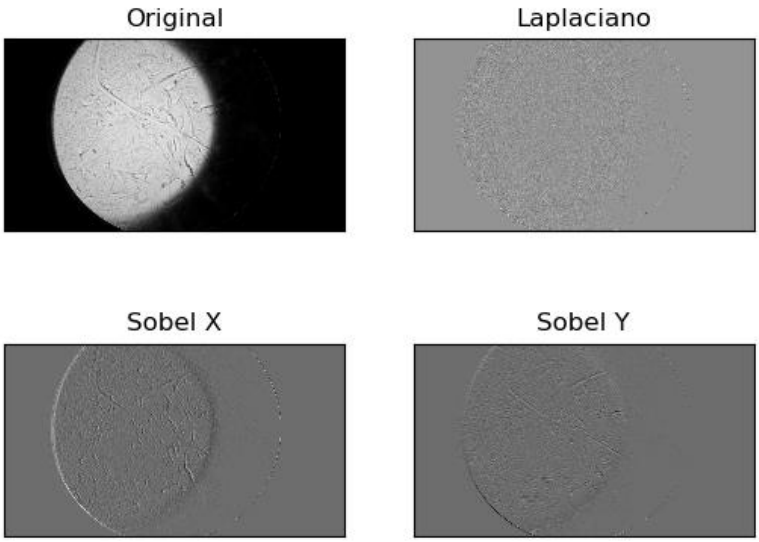


Figure 23 - Filters applied to a sample completely out of focus

d) This image also shows a sample out of focus and with the camera displaced to the right:

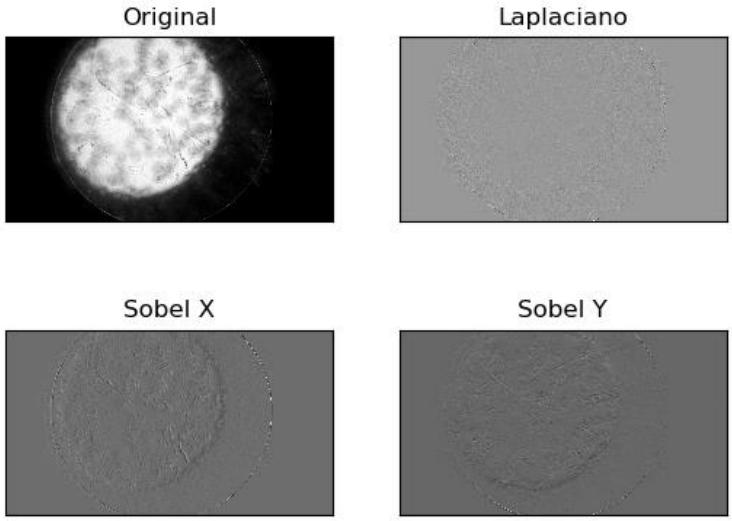


Figure 24 - Filters applied to a sample out of focus and with the camera displaced to the right

Those images were evaluated with the functions previously presented. However, the resultant matrices have a dimension of 1080x1920, which is difficult to compare, value per value, and obtain a determinant result. Furthermore, Python IDLE[6] could not even print the whole matrix.

[6] IDLE is a development environment for Python.

Therefore, in order to make this start easier, in the first place it was considered to compare the maximum value of the Laplacian. This was supposed to reduce the computing time because there is only one resultant matrix that does not require of additional calculations. As the only part of the image that can present a variation on the gradient is the sample (the frame of the sample is the objective and is plain black, which makes that for the entire frame, the values of the matrix are 0 or almost 0), it will be taken the maximum value of the whole matrix. This should have made that for unfocused images; the maximum value was considerably lower than for focused images. For example, for the previous images, the maximum values of the Laplacian were the following ones:

a) Figure 15: 2072266
b) Figure 16: 1298
c) Figure 17: 2072812
d) Figure 18: 78187

It can be seen that unfocused images present a considerably lower maximum in their Laplacian. However, the image that was completely out of focus, presents a similar value than the focused one. This only demonstrates that this criterion is not valid to determine if the picture is focused or not.

Therefore, based on the article written by Said David Petruz Arroyo and Héctor Reneé Ibáñez Grandas about acquisition of images using an optical microscope [12], a different analysis will be performed.

In this article, they propose the usage of different parameters to evaluate whether the images are focused or not. They reach the conclusion that the best method to determine if the image is focused is using the modified variance of the gradient.

Thus, following this article, the next step is to calculate the modified variance of the Laplacian. To calculate the variance it is used the following function:

```
Variance=numpy.var(matrix)
```

Where numpy is a Python extension that supports operations with matrices and vectors. This function basically performs the following calculation [28]:

$$Var(x) = \frac{\sum_1^n (x_i - \bar{X})^2}{n}$$

Where x represents each value of the matrix, $\bar{X}$ is the media of the values and n is the number of elements.

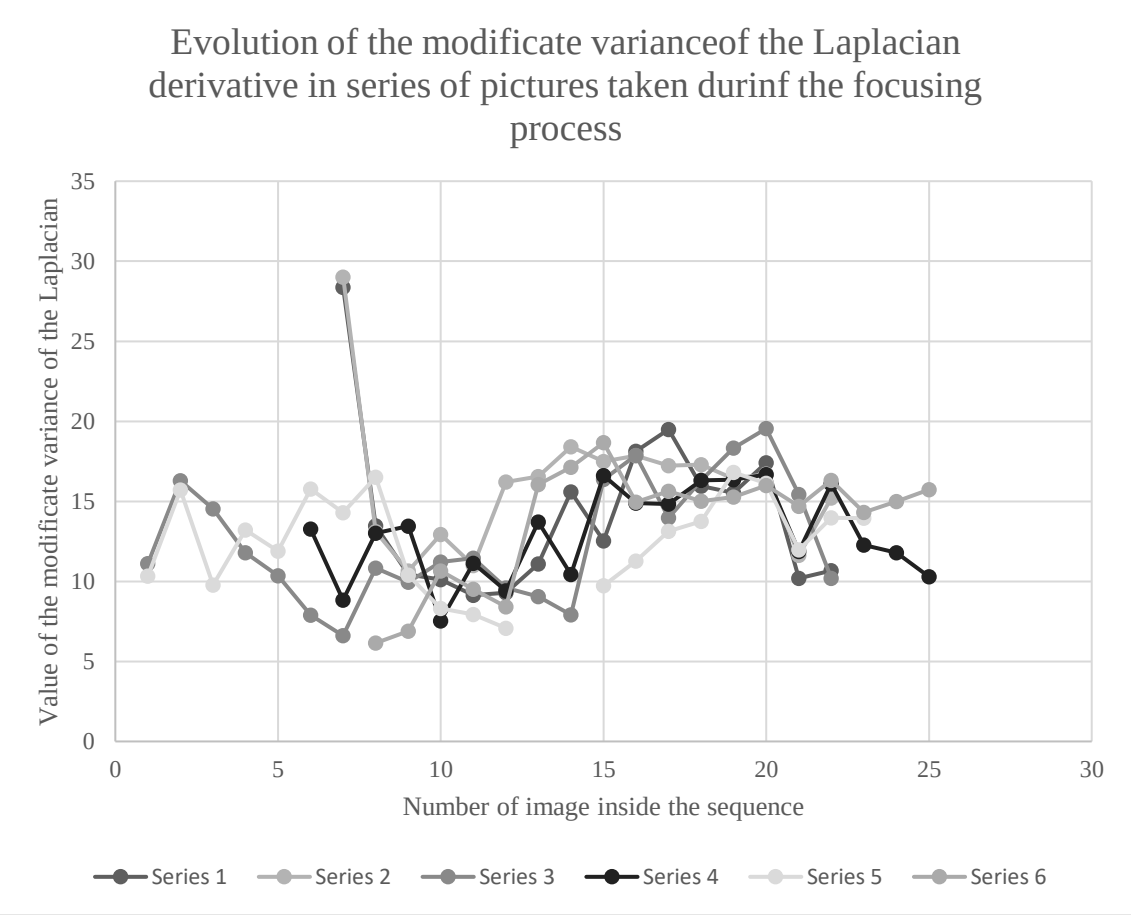
Then, the modified variance is calculated using the following expression:

$$Modified\ Var(x) = \sqrt{Var(x)^2 * \frac{n}{n-1}}$$

In this case, there is no function to calculate this so it is obtained as a series of arithmetic operations. It must be said though, that calculating the modified variance, with the resolution of the cameras used to develop this project has no noticeable effect because the number of elements is huge. However, this will be kept just in case a worse camera was used with the microscope in the future.

Finally, with the series of images captured to study the focusing of the images, the values in Table 8 are obtained. With a naked eye, one can see that the values do not represent a function from where a rule can be obtained. Moreover, if the values are graphically represented, this can be corroborated.

Table 2 - Graphic representation of the modified variance of the Laplacian in series of images taken during the focusing process



It is noticeable that values do not have a significant variation or peak in the point where the images are focused (around 20), actually, the values present a slight variation but constantly fluctuate without giving an apparent result.

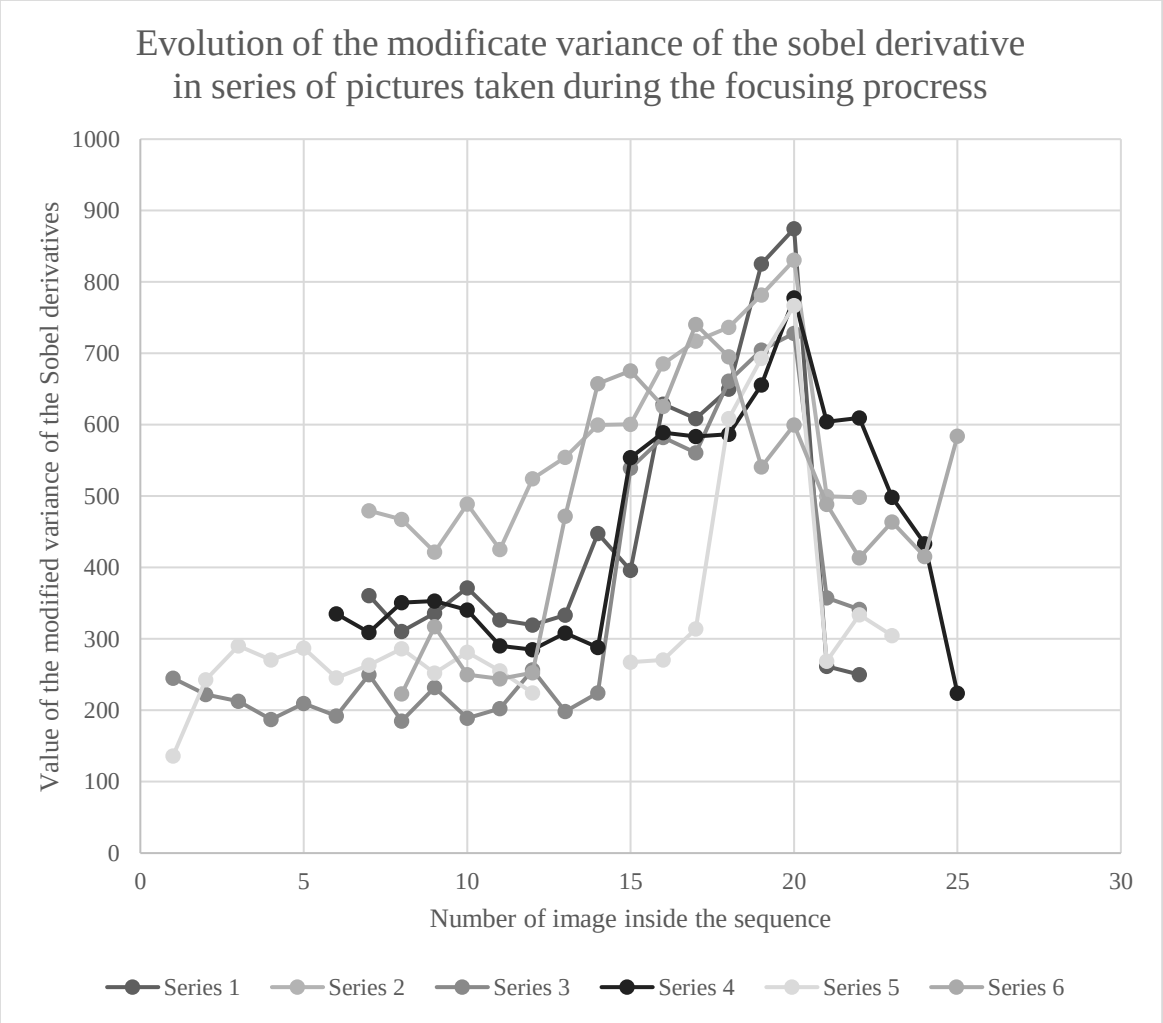
Having seen that, another point of view is considered. Even though it would require a higher computation time, it uses the Sobel derivatives. In this case, the function proposed is the following one [12]:

$$Gradient = |G_x(x,y)| + |G_y(x,y)|$$

One can notice that the gradient would actually be the square root of the sum of both terms elevated to two. However, this function was tested in first place because reducing the number of operations performed with the matrices will automatically reduce computational time and reduce the risk of an error.

Then, with the previous formula, the variance is calculated from the gradient and then, the modified variance. Finally, the data presented in Table 7 is obtained, and again, the results can be seen graphically presented in the following table.

Table 3 - Graphic representation of the modified variance of the Sobel Derivatives in series of images taken during the focusing process



Whereas in the first plot no significant difference could be seen, in this case, it is clear that the values obtained have a clear maximum around the image tagged with the number 20, which is actually the focused one.

Therefore, to focus the image, the modified variance of the gradient must be calculated, and, with these values, the microscope can be guided by searching the maximum of the series.

Testing of the validity of this method

In order to test if this method that has been implemented and which is the best reference value to determine whether an image is focused or not, the following code was implemented:

Script to test if an image is focused or not:

```
import cv2
import numpy as np
import math
```

```
#1. importing the tested image
img=cv2.imread('imagename.jpg',0)
height,width=img.shape[:2]

#2. calculation of the sobel derivatives
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=3)
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=3)
sobel=np.abs(sobelx)+np.abs(sobely)

#3. calculation of the variance of the sobel
var1=np.var(sobel)

#4. calculation of the modified variance of the sobel
n=height*width
mod1=(var1**2)*(n/(n-1))
modf=math.sqrt(mod1)

#5. Comparing with the reference value
refval= #reference focus value

if mod1>=refval:
    print("The sample is focused, the image value is %d." %mod1)
else:
    print("The sample is not focused, the image value is %d."
%mod1)
```

This code allowed the user to analyse the captured image and print the value the modified variance of the gradient and a message saying if the image was focused or not.

After many test runs, which can be seen in Table 9 the final reference value to decide if the image was focused was established in the following value:

$$Reference\ focus\ value = 370$$

This value has been chosen following this series of considerations:

a. It is clear that when the sample is too far from the objective the value of the modified variance never exceeds 300. So, this is a safe value to always reach at least the point where the sample is focused.
b. When the sample is too close to the objective (ergo, the focus point has been passed), even though it never reaches 400, the value can exceed 300.

c. When the image is focused, the value usually passes 300. However, in this case two problems show.
    a. If the images are blurry, the values can be so low that never reach the reference value. In this case, the most appropriate would be to repeat the process because it can be that the image has not been able to be focused or it can be that the images are blurry, which would make them useless to analyse.
    b. If the images are a bit out of focus, this is that they may need 0.1 mm to be adjusted, maybe the values are a bit low or not, also depends on the quality of the image. So, in this case the two options can be presented: the images are good, which is acceptable because they can be analysed anyway or the images are a bit blurry and are discarded, which simply would imply a time loss because the analysis would have to be repeated.
    c. If the images are perfectly focused their value always surpass 400.

Because of these reasons, the value 370 chosen. This value would mark as good zero images that are not but would discard eight images that could be analysed. This implies an error of a 5% in the precision of the microscope. Taking into account that this is mainly produced because the structure of the microscope is unstable and that, as it has been said, will not have any negative effect in the microscope performance but to repeat some of the analysis it can be accepted.

Nevertheless, this will not be the only required condition to consider a point the focus of the sample, it will be also based in the reference value of the other images that have been taken.

However, it must be added that this value will have to be revised every time the structure of the microscope is modified to adjust the precision. It has been seen that for a completely stable microscope, the values are considerably increased and the only reason that they are lower is the constant shakiness of the microscope. It is recommended that is also adapted to every phone that is used with the microscope to avoid imprecisions.

Chapter 4

Displacement of the sample

In this section it will explained how it has been automated the displacement of the sample in both directions: vertical (respect to the objective) and horizontal (changing the portion of sample that was seen in the camera).

Vertical displacement of the sample

The vertical displacement of the sample is thought to remain as it was at the start of this project, except it will be automatic. In this section, it will be explained how this mechanism works and how it has been automatized.

Vertical displacement mechanism

The vertical displacement of the sample uses the following mechanism:

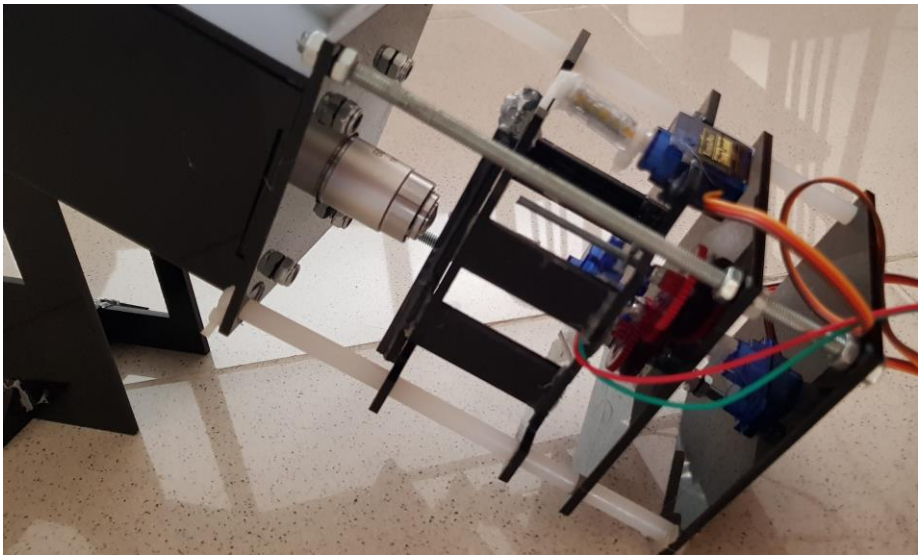


Figure 25 - Moving parts of the microscope

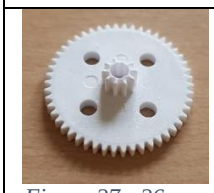
The mechanism has the following parts. It starts with the servomotor, which is driven by the Raspberry Pi and that is fixed to the last platform of the microscope. Previously, instead of the servomotor, it had a wingnut that had to be adjusted manually. The servomotor's torque is transmitted to the middle axis of the immediately superior platform. The axis is fixed to the servomotor using a silicone tube. A silicon tube is used instead of any rigid part that could be 3D

printed or any rigid union to prevent this part of breaking in case there was any misadjust. So, this axis has a fix gear that is at the same time transmitting its movement to two more gears.

These gears have the principal function of reducing the speed of the movement that is transmitted.

They present the following characteristics:

Table 4 - Characteristics of the gears used

Gear	Number of teeth	External diameter	Diameter of the axis	Height of the part that is in contact with the other gears	Total height of the gear	Module
 *Figure 26- 15 mm gear*	26	15 mm	3 mm	2 mm	6 mm	0.5
 *Figure 27 - 26 mm gear*	50	26 mm	3 mm	2 mm	6 mm	0.5

Therefore, the relation of reduction of these gears can be calculated using the following expression [29]:

$$\tau = \frac{Z_1}{Z_2}$$

Where τ represents the relation of reduction and Z is the number of teeth of each gear. From this relation, it can also be extracted the relation between the angular velocities of the gears (which is the inverse of this one calculated) and the relation between the torques they are transmitting (which are proportional to their respective angular velocities).

So, the relation in this case is the following one:

$$\tau = \frac{Z_1}{Z_2} = \frac{26}{50} = 0.52$$

It can be seen that is not really high reduction value, but due to the lack of space in the microscope the gears cannot be made much bigger.

Then, these bigger gears are fixed to two more M3 threaded rods, which are connected to the part of the microscope that contains the sample. This part is connected to the threaded rods using M3 nuts, which makes that when the servomotor is on, the bars start turning and this part, with the sample, moves up or down with it.

The result of this movement is a final maximum value of the vertical velocity of the sample of:

$$v_z = 0.43\ mm/s$$

Considering that for a M3 nut, the thread pitch's is 0.5 mm, the speed of the servomotors is 60º/0.1s and the reduction relation is 0.52.

Automation of the vertical displacement mechanism

Therefore, once the mechanism has been explained, it will be proceeded to explain its automation.

The first part of the automation consisted on the addition of the servomotor. This part was developed in collaboration with Joana Aina Martorell, to whom was assigned the redesign of the microscope parts [1].

Joana added the last piece, which oversaw maintaining the servomotor fixed in its place and prevent it from spinning over its own axis instead of transmitting the spin to the axis. It was also incorporated the silicone tube which was used to fix the axes one to another.

Once the system was ready, its automation could be performed. The first step was to fix an initial and a final point. This part could have been much easier if there were added position sensors, but this would have incremented the cost of the microscope and required an extremely high level of precision in the assembly, and as it was not compulsory, it was avoided.

In this case, it was easy to determine an initial position, which had to be considerably separated from the objective. This way, it was prevented that the user of the microscope could damage the lens while changing the samples. However, a final position could not be fixed because focusing the sample is expected to be an iterative process, which needs to leave behind the final position while taking pictures to return to it later once it was seen that this was the focusing point.

Therefore, this process was thought to keep a reference value to its start and modify it as the displacements were performed, so in the end, the mobile parts could be returned to their initial position. This can be seen in the Main program section of the Annex 1: Code used with the Raspberry Pi and the Raspberry Pi camera.

To implement this code, it has been compulsory to calculate the duty cycles of the servomotors [23]. Servomotors usually work in a range of 2.5 to 12.5% of duty cycle [30]; where the lowest duty cycle represents the highest velocity in one direction and 12.5 the highest velocity in the opposite direction. Thus, it can be deduced that the middle value (7.5%) will be the one that keeps the servomotor resting. However, although this value is exact for some servomotors, for others it is not so it needs to be tested for every servomotor due to possible maladjustments.

Therefore, the reference values that will be used in this part and with a unique servomotor that is adjusted to them are the following ones:

a. 4.5%: this value will be used to move down the sample.
b. 7.5%: this value will be used when the sample must be resting.
c. 10.5%: this value will be used to move up the sample.

Once the values were defined, a study of how much time the sample needed to be moved in order to reach the focused position was needed in order the keep the initial position wide enough to prevent the user from damaging the objective while changing the samples.

The initial position was defined as the lowest position possible. In this position, the sample could not be further from the objective because the M3 threaded rods that displace it are in contact with the piece that contains the sample:

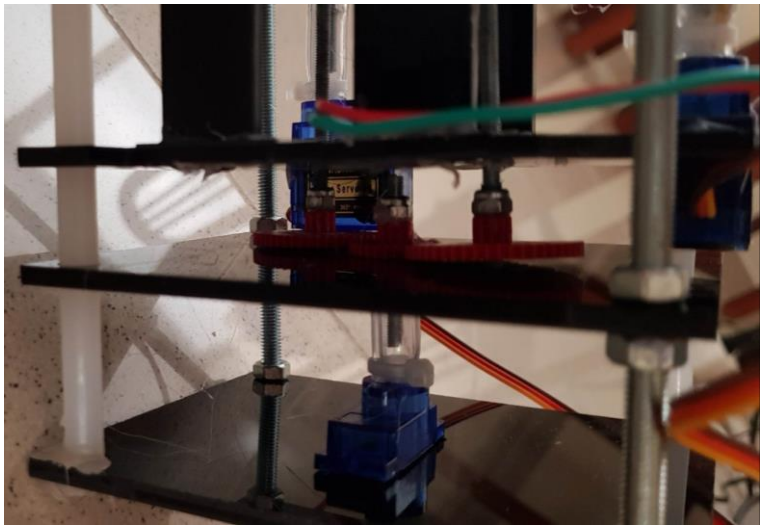
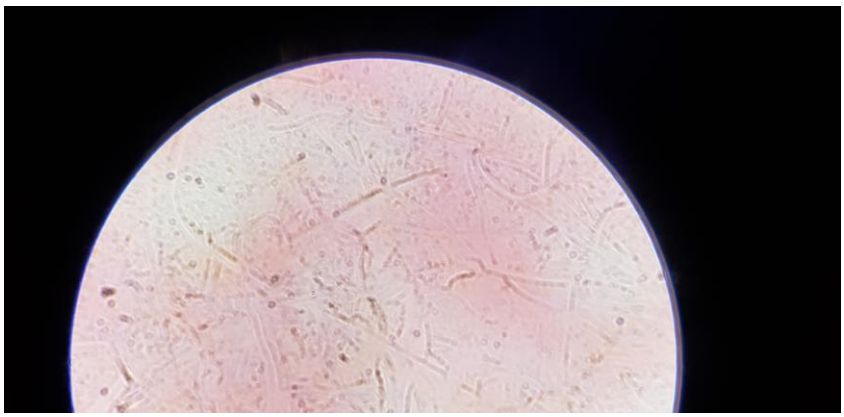


Figure 28 - The gears and the moving part in the initial position

Then, a standard time of 16 seconds was stablished to move up the sample so it could reach a position where the sample could start to be seen but was not focused to prevent the focus point from being skipped. However, this value depends completely on the distance that the manufacturer leaves between the moving parts and the lens, so it will have to be adjusted for every microscope.

Then, it was also evaluated how much time it was needed to acquire enough precision in the movements of the focusing process. This process started with a time lapse of 1 second, which was clearly much higher that it would be. That time were reduced by half in each iteration until it reached the value of 0.125 seconds. With this time value, the movements had enough precision to always reach the focus point of the sample and no present a considerable variation from one image to the following one. This can be seen in the following image sequence, which was taken moving up the sample in intervals of 0.125 seconds.



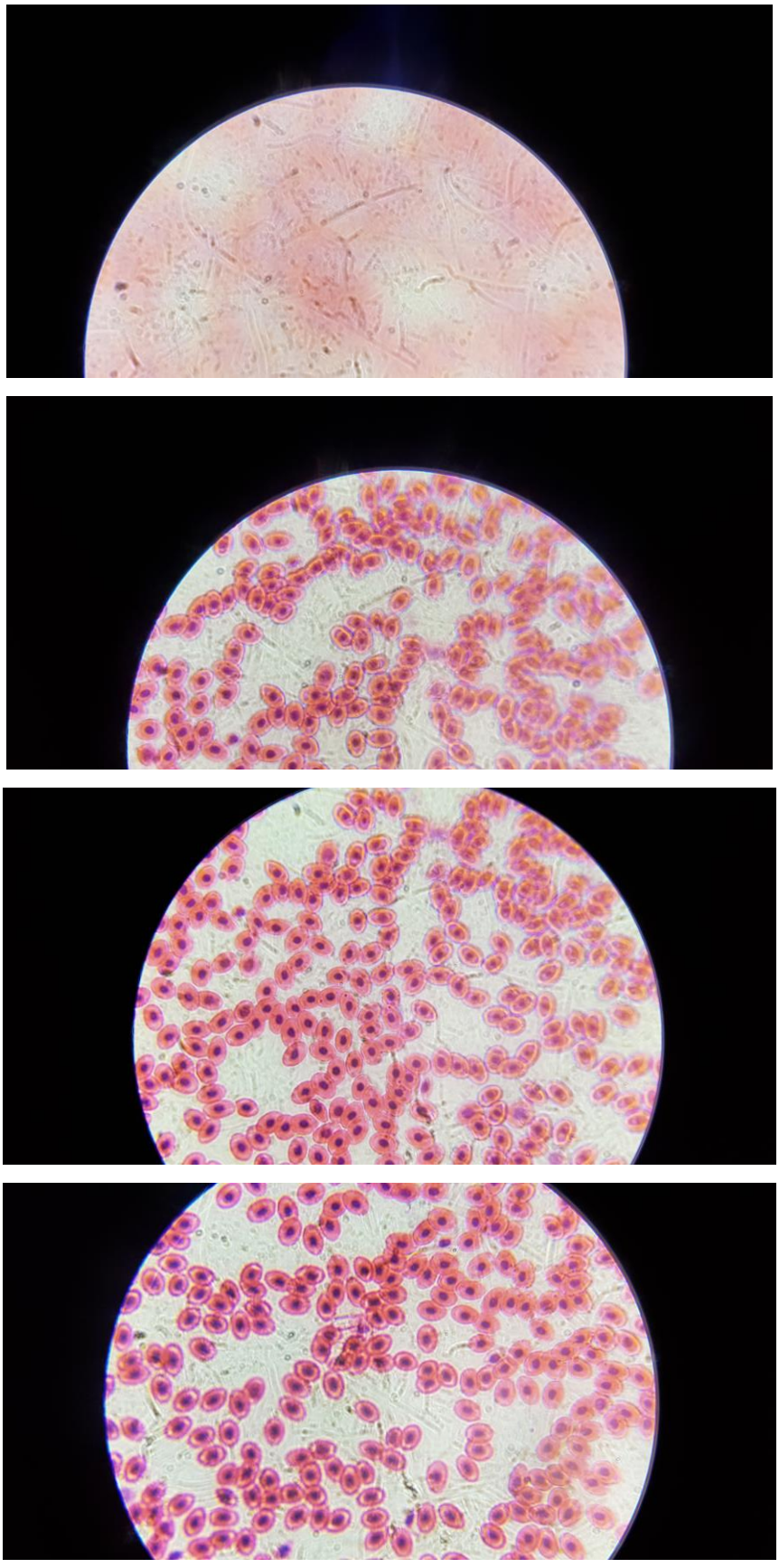


Figure 29 - Sequence of images extracted from the sequence 6 of the Table 7

It is clear that this interval gives the microscope an appropriate precision. However, it will incorporate the option of moving 0.0625 s just in case more precision was required in any moment.

Finally, the sample is moved down the same exact time it has been moved up previously in order to keep the initial position.

All this movements can be seen again in the Main program of the Annex 1: Code used with the Raspberry Pi and the Raspberry Pi camera.

Horizontal displacement of the sample

This part of the project has been developed in collaboration with Joana Aina Martorell, who was modifying the structure of the microscope and designing the new moving system. If the reader wanted to see the whole process of this part of the system being redesigned, one should read about this in her project [1].

Initial approximation

In the section Initial approximation, the iteration of the microscope where those projects were started was shown. Initially, it was tried to make it work with the proposed motors, but it was clear that the final system could not work like that. It presented the following problems:

a. In order to make it work, the motors had to be placed in such inconvenient places. This placement made the microscope a lot less portable and fragile.
b. The placement of the motors itself was a problem because in some cases they were much heavier than the methacrylate pieces, joint with silicone could resist. In addition, the position into which they were positioned created the additional requirement of adding a support for them because if they were not added, the motors fell and interfere in the microscope correct working.

These are the main reasons why this system was redone. In order to start the designing process again, some ideas were proposed to see which one of them was more suitable for the required movement.

These ideas are presented in the following enumeration with the corresponding reasons why they were chosen or not.

a. First, it was proposed a solution that involved the motors connected to the sample through bars with joints. That should allow that the turning of the motors were transformed into a straight movement of the sample, and with two motors, the sample might have been able to move in the two directions. This could have been an option if it was not because there is no space available to place the bars and allow them to cover the entire needed trajectory.
b. Second, it was proposed a solution that involved designing lateral joints for the sample, so it had two holes in the extremes. These holes were supposed to be one, fixed to the immediately inferior platform and, the other, controlled with a servomotor. This would allow a circular movement of the sample using a turning radius considerably small and also would require of only one servomotor.
   The main problem of this alternative is that this design would only allow a fix trajectory of the sample. That would mean that in the case the sample was not exactly

the same shape or in the same position than the one it was calculated for, it would not be useful.

c. Another alternative was a plot clock[7] mechanism. This mechanism would allow an extremely precise movement but has the same main disadvantage that the first one mentioned. The arms of the plot clock need to be big enough to allow it to make all the movement and there is no space to place these arms.
d. Thus, the following alternative analysed consisted in adding gears with racks that allowed the movement in the two directions. This would only require the design of two pieces that has guides to move one over the other and racks to be moved using the servomotors. However, this idea had to be discarded too because the properties of the methacrylate did not allow this type of movement and because the servomotors needed to be displaced too and this was an inconvenience.
e. Finally, the last idea that was developed consisted in a slight variation of the first one. Instead of using long bars to move the sample, it was proposed to use a single arm in one side of the sample that moved it up and down with a 90º movement of the servomotor. At the same time, in the opposite axis, it would be a connecting rod. This rod would move the sample to the left or to the right depending on its position.
This was the chosen alternative because it allowed the desired precision and was, at the same time, a simple mechanism.

This chosen alternative, presented the following final aspect:

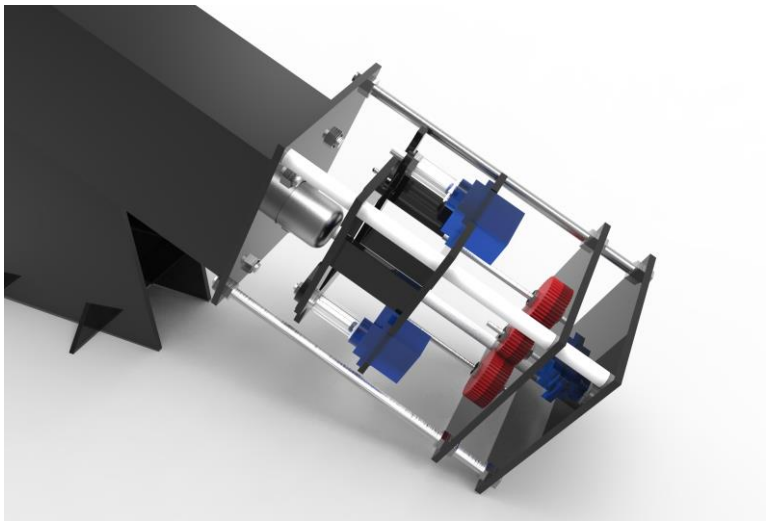


Figure 30 - Final design of the AiScope moving part. Author: Joana Aina Martorell [1]

However, neither this is the valid solution. This solution was performed without taking into account that for the selected optical, the sample needed to be in contact with the lens in order to focus the sample. Thus, with the designed system it was neither advisable nor possible to move the sample in the focused position because the contact with the lens and the construction itself made the displacement impossible.

Nevertheless, this problem did not made the process impossible. Due to this inconvenience, the sample could not be moved in the plane while it was focused but it could be unfocused, displaced and refocused. This, obviously, will make the automation process considerably inefficient and slow but will work for the first automation iteration.

[7] A plotclock is a robotic clock that plots time. It is designed to subject a pen and write down the time driven with an Arduino board or something similar.

It is important to remark that this is the first iteration of the automatic microscope and that not all the problems or usage difficulties could be detected and solved in the time this project and the redesign project [1] were developed.

Programming of the proposed solution

In order to program the proposed solution, the first step was to add the servomotors. To add the, two rectangular spaces were made in the base of the moving part.

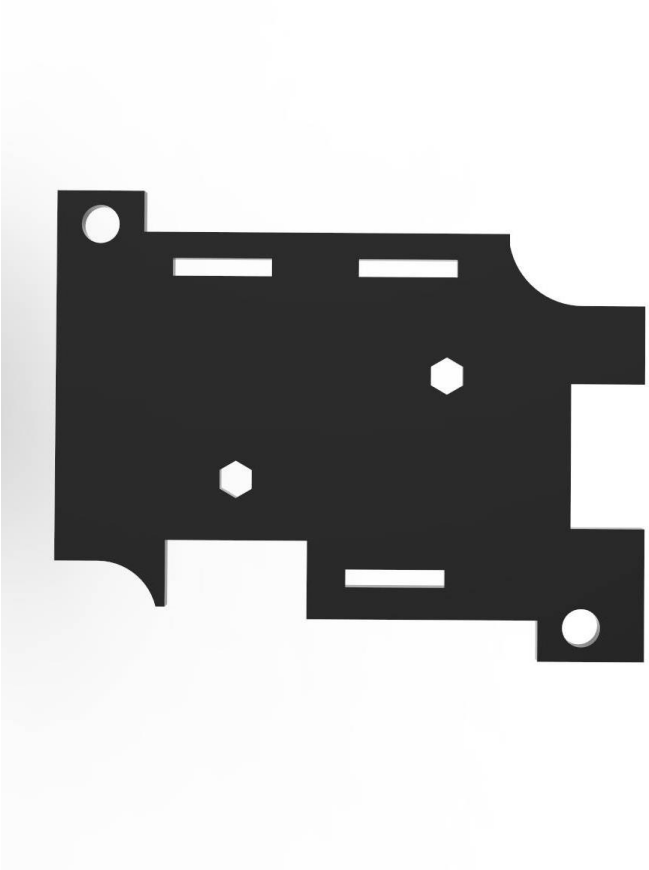


Figure 31 - Part where the servomotors are embedded. The two spaces can be seen in the right of the piece and in the left bottom corner. Author: Joana Aina Martorell [1]

This way, the servomotors had the exact space to be embedded in the microscope and did not constituted an outgoing part that could be a problem in the future. This way, it also could be used the small holes that the servomotors have in both sides to fix them if it was necessary.

This way, the system has the following aspect with the servomotors on it:

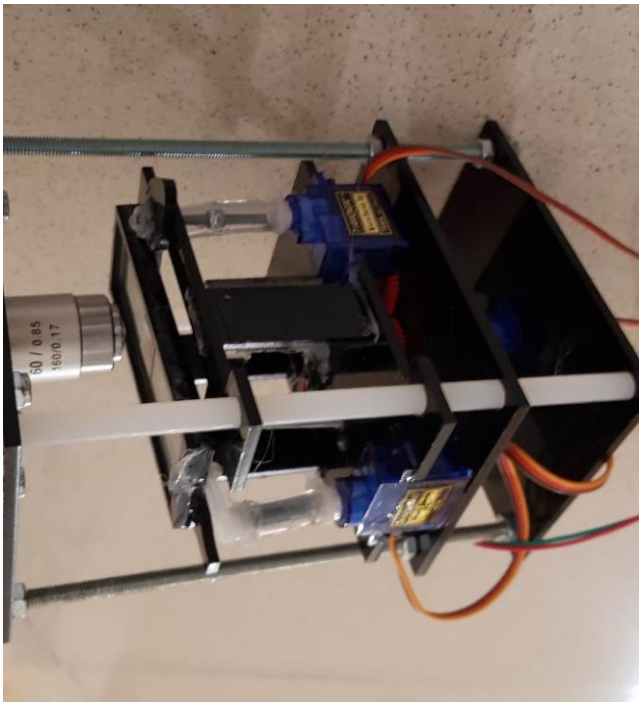


Figure 32 - Final aspect of the moving part with the two servomotors

With this system, it needs to be achieved that the camera makes a complete trail throughout the sample. To implement this, it has been performed a study of the movement of the sample depending on the movement of the two moving parts: the crank and the connecting rod.

The crank made the sample move in a semi-circular motion that basically displaced the centre of the sample along the radius of the crank. This allowed the displacement mainly in one direction (which will be considered y-axis direction in the codes). On the other hand, the rod moved the sample to the left from its original position, so for every position got with the crank, there were two positions of the rod. This movement of the connecting rod will be considered in the x-axis in the codes.

Thus, considering this, on the one hand, the servomotor that controls the crank will not be a 360º servomotor, as the others, but a 180º servomotor. This decision has been made because the purpose of this movement is first, to keep the sample under the objective, which cannot be accomplished if the samples is rotated with a higher radius and second, to move in a precise way, which can be better achieved with a 180º servo.

Then, the movement options of the servomotor were calculated. The servomotor was expected to turn inside the circular guide that has been made in the piece that contains the sample. This is exact because the guide has the shape of a semicircle. This guide can be seen on the right of the piece in the following image:

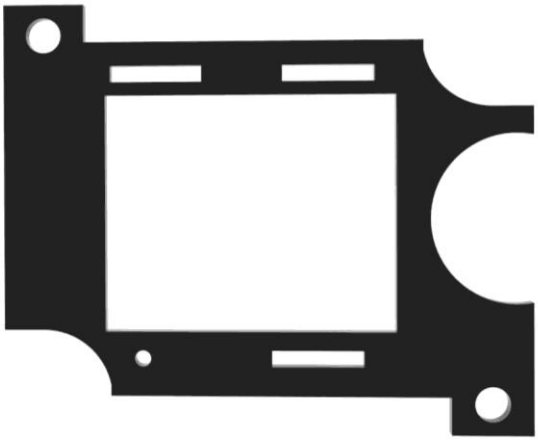


Figure 33 - Superior piece of the moving part. The guide for the crank can be seen on the right. Author: Joana Aina Martorell

Therefore, the servomotor, as usual, works in a 10% of the Duty Cycle window, which is due between 2 and 12%. This way, it can be deduced that each 1% will represent a degree variation of 18º. For this reason, the step between one position and another will be of a 0.5% of Duty Cycle. With this increment, there will be 20 different positions analysed, which is enough to cover a great part of the sample.

On the other hand, the servomotor that drives the connecting rod will be a continuous turning servomotor. This servomotor will mainly control the position of the rod, making it to be in its shortest or longest part touching the sample depending on the desired position. In order to do this, in only needs to be calculated the time and speed that the servomotor requires to do this action.

Therefore, knowing that the resting Duty Cycle of the servo is of 7.5%, the speed will be fixed with a Duty Cycle of 7.75%, so it is moved slowly. With this speed, the time to complete half a turn is of 0.58 seconds.

As before, this time is the considered with the concrete servomotor used to build the prototype. However, when a new servomotor is used or when a new AiScope is built this value needs to be tested.

So, to get this process to be performed the two servomotors have to be programmed the following way:

```
#first step will be to import the libraries needed to execute the
#program
from time import sleep
import RPi.GPIO as gpio


#set up of the pin mode (this will determine the naming of the
pins)
gpio.setmode(gpio.BCM)
```

```
#set up of the pins that will be used as the servomotor output
gpio.setup(20,gpio.OUT)
gpio.setup(21,gpio.OUT)
#set up of the PWM frequency: 50 Hz
x = gpio.PWM(20,50)
y = gpio.PWM(21,50)
#initial position of the servomotor
#position x (the rod) is initialised in 0 (resitng)
x.start(0)
#position y (the crank) is initialised on the right
y.start(2.5)

#Then the sequence of movements is initialised

p=2.5
while p<12.5:
    #first the crank is moved
    y.ChangeDutyCycle(p)
    #here the first picture would be taken
    #then the rod is half-turned to get the other position
    x.ChangeDutyCycle(7.75)
    sleep(0.58)
    x.ChangeDutyCycle(0)
    #here the second picture would be taken
    #and then the rod is returned to the other side
    x.ChangeDutyCycle(7.75)
    sleep(0.58)
    x.ChangeDutyCycle(0)
    p=p+0.5

#To finish the script the GPIO is closed
gpio.cleanup()
#And would make a total of 40 pictures of the sample
```

This code will be incorporated in the main codes in order to perform these movements when they are required.

The last thing that needs to be mentions about this movement is the placement of the connecting rod. The rod is connected to the servomotors with a M3 threaded rod that goes through the small hole that can be seen in the Figure 27 of the piece that is right below the sample. The hole is located on the left top corner of the piece. This way, it can move the free side of the sample.

Final algorithm and electronic assembly

The final algorithm follows the structure shown in the following flux diagram:

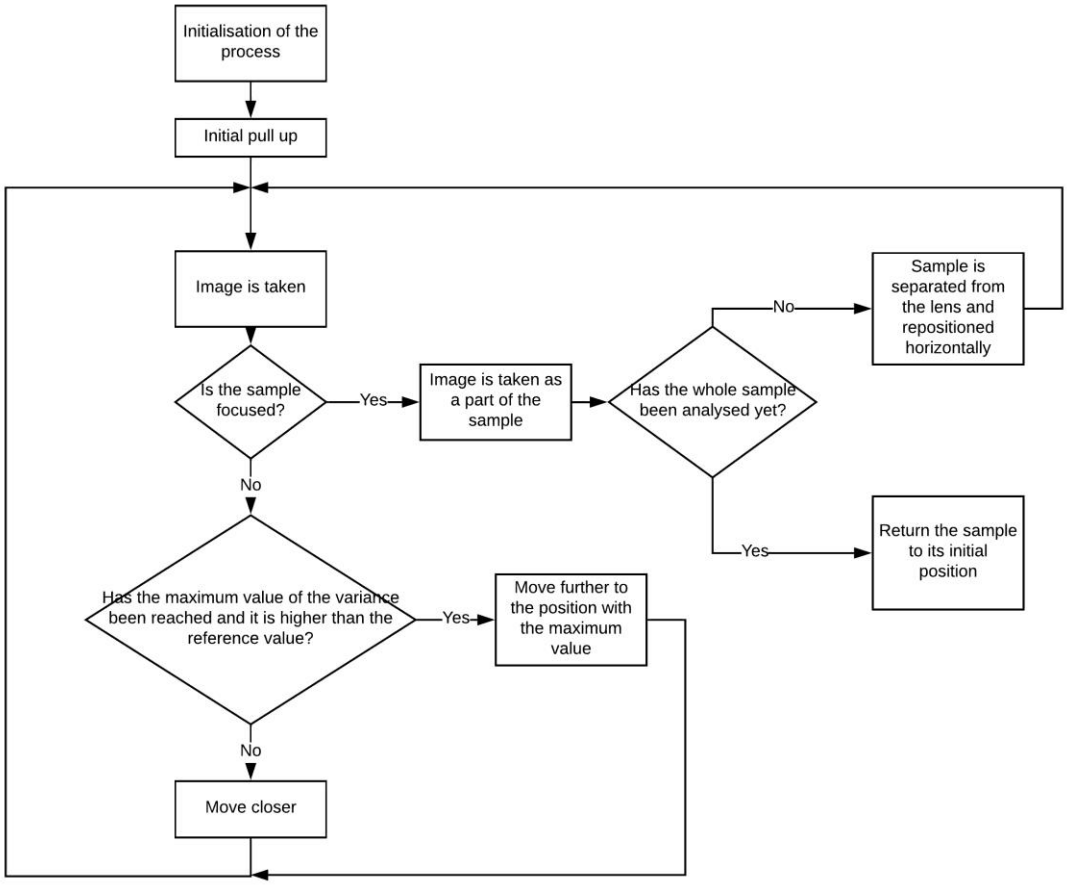


Figure 34 - Flux diagram of the main algorithm

The code of this algorithm can be found in the section Main program using the Raspberry Pi camera. It can be seen that the algorithm is based in analysing the image in the microscope repeatedly until the sample is focused and then it takes the definitive picture. This process is repeated until the sample has been analysed throughout.

The focus of the sample is evaluated with the reference value and it determines the vertical position of the sample. However, this vertical position is not constant due to the high precision that is required. A single movement of the microscope can make the microscope lose its focus, so that is the main reason because the sample is refocused before taking every picture.

And to implement this algorithm, the following electronic assembly is required.

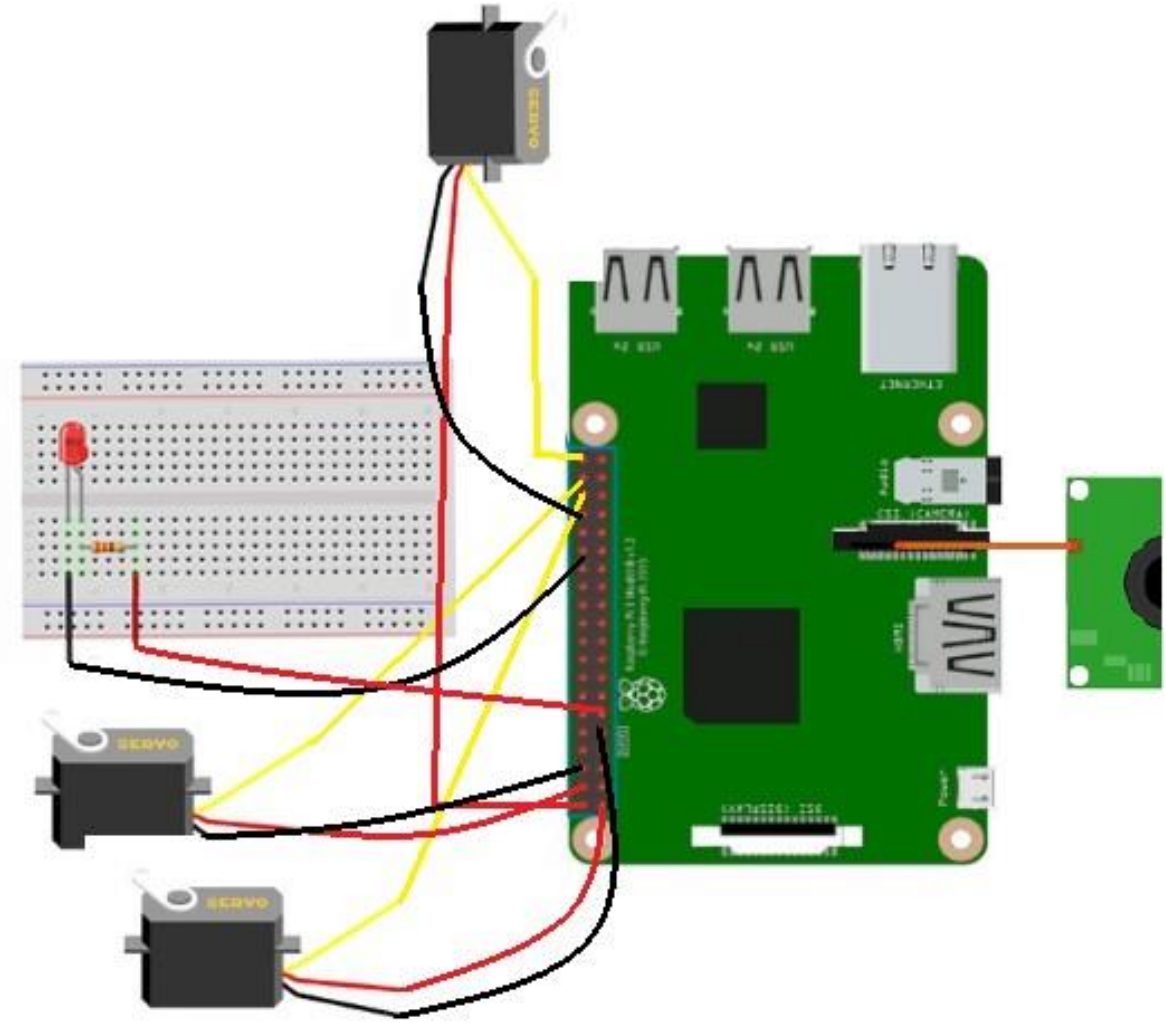


Figure 35 - Electronic assembly using the Raspberry Pi camera

Notice that the three servomotors are connected at the same time to one ground, one voltage input and one GPIO. The GPIO pins are the numbers 16, 20 and 21. This will be seen again in the Main program using the Raspberry Pi camera. The camera is just connected to the camera input and finally, the led is connected to a ground and a GPIO pin (number 17). It must be said that in the previous figure it is assembled using a protoboard but in the final microscope the protoboard will disappear. It was just used to bring stability to the assembly. Furthermore, it is also compulsory to remind the reader that the led is connected with a GPIO to control if it is on or off but if one wanted to keep it turned on all the time, it could be connected to a 3.3V voltage input instead of the GPIO.

Chapter 4

Improvements

Changing the Raspberry Pi camera for the mobile phone camera

This section will recompile the different alternatives that were evaluated to connect the mobile phone with the Raspberry Pi and the respective arguments for and against.

The three alternatives that were evaluated are:

a) USB connection
b) Wi-Fi connection
c) Bluetooth connection

They will be presented in order with a brief explanation.

Presentation of alternatives

USB Connection

Connection via USB is easy, but it can suppose a problem when it comes to compatibility of the connectors. Using this connection would imply that all the phones that are used to take the samples have the same input. This means total incompatibility for android/iOS phones because they have different inputs.

This can be solved if, as it is expected, one unique phone is used with every AiScope. Thus, one connector can be left with the Raspberry Pi so the user only needs to plug in that phone before using it.

If was not possible, another solution would be to leave the Raspberry Pi open to the user so they can connect their own cable but this presents two additional problems:

a) The Raspberry Pi would be uncovered to untrained people who can break it or make that other connections fail. Also leave the Raspberry Pi uncovered would leave it in contact with aggressive environments that can affect to its proper function (specially in humid environments)
b) This would require that people who wanted to use the microscope have to carry their charger cable with them, which can present a problem in case they did not.

So, ideally it would be recommended to use only model of phone with the AiScope and that its connector would be incorporated to the system in its construction, so untrained people do not need to touch it.

Bluetooth Connection

Bluetooth connection presents the main advantage that is wireless but at the same time does not require of internet connection, so if the microscope wanted to be used in an isolated place where internet connection was not available it would be possible.

However, Bluetooth connection does present several incompatibilities between devices that are difficult to be solved in situ.

Moreover, using Bluetooth connection would require of the designing and programming of an application that received data from the Raspberry Pi and sent data (this time including not only text but also images) from the phone to the Raspberry Pi. This would include in this project a new programming language: java; and the knowledge of how to program an app (both the layout and the contents). Java programming language is not a competence of this project specifically because it has been included neither the required time to do this in the planning nor the time to learn java or app programming.

It must be added that some simple software to create apps without programming with java have been tested but, in general, they do not include the option of sharing images via Bluetooth because they are not that complex.

Wi-Fi connection

Wi-Fi connection, as it has already been commented, presents the main inconvenient of not being available in every place, especially in isolated places. Using Wi-Fi will turn into a requirement the availability of mobile data.

The connection between the mobile phone and the Raspberry Pi is easy. With the two devices connected to the same Wi-Fi network, one can access to the other using the IP address. The IP address of the Raspberry Pi can be obtained using the following command:

```
sudo iconfig
```

With that command, the Raspberry Pi prints in the command window the IP address.

Moreover, to use the camera directly with the Raspberry Pi an App would also be required, but as it was commented in the previous section, it will not be implemented for now, so an alternative had to be used.

Final options that were considered

Wi-Fi alternative

This alternative will consist in using an IP Camera App to send images from the mobile phone directly to the Raspberry Pi. This App works livestreaming the image of the camera into an IP address, which can be accessed by any device, connected into the same Wi-Fi network. One benefit of this method is that if the microscope is located in a place where there is no Wi-Fi, one can use their mobile phone as a hotspot to share mobile data and they are directly connected into the same network.

In this case, the App that has been tested is called IP Phone Camera. This App basically broadcasts the image that is seen through the mobile camera into the Wi-Fi network. The App presents the following layout:

Figure 36- Layout of the IP Camera App

Moreover, the website where the computer or, in this case, the Raspberry Pi is connected has the following layout:

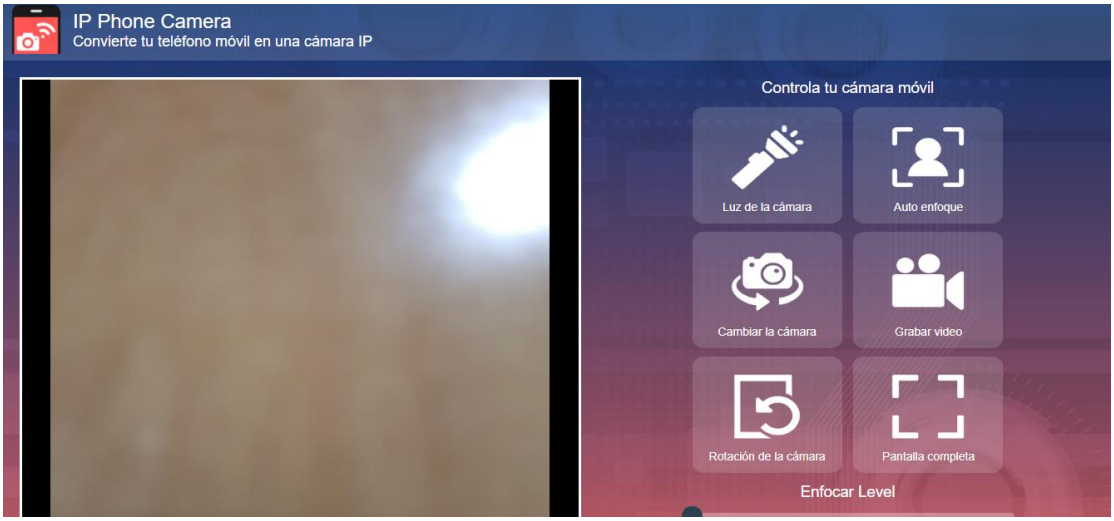


Figure 37 - Layout of an IP Camera App in the computer

However, this option presented three main disadvantages:

a. Even though it could be used via mobile data, if there were no mobile data available the microscope could not be used.
b. The process of livestreaming the images incredibly reduced the quality and resolution of the images that could be obtained even when using a Wi-Fi network. That would difficult the process of focusing, especially if the microscope had to use mobile data and they were insufficient.
c. These apps protect their images so one cannot do anything with the Livestreaming that is doing. Because of this, it is difficult to obtain them using a computer and hardly impossible for the Raspberry Pi using a simple coding.

USB + Bluetooth alternative

This alternative involves the use of both, USB and Bluetooth. These, initially alternatives, have been proposed to be used together due to the lack of transferring images possibilities of the software used to develop an App.

This solution involves the use of Bluetooth to allow the communication between the mobile phone and the Raspberry Pi and the USB to transfer from the phone to the Raspberry Pi the images.

In order to use this alternative, the following will be proposed:

a. To solve the problem of the compatibility between the phone cables and the different models, one universal charger will be installed, and the user will be advised. That way, if the user knows that their phone is not compatible, they can bring their own cable or, preferably, an adapter.
b. The phone will be paired to the Raspberry Pi in the moment of the preparation of the previous. That way, the used will know from the start if there is any incompatibility between their phone and the Raspberry Pi and try to solve it.

This option, apart from the disadvantages presented with the different alternatives, did not present any major inconvenience. Thus, this alternative will be chosen to be developed.

Implementation of the chosen solution

This solution involved many parts: programming the Bluetooth connection in Python, creating an App to allow the Bluetooth connection, programming the file sharing via USB between the Raspberry Pi and the mobile phone.

Bluetooth connection in python

The Bluetooth connection in python does not require of many steps. It is simplified with the Bluetooth library, which allows a simple connection via RFCOMM.

Using this library, the steps to connect the Raspberry Pi as the server are the following ones. First, one need to stablish the host and the port for the communication. The host does not actually need to be defined, because the connection can be left open to any device that wanted to be connected in that moment. Then, the server needs to be created using the BluetoothSocket and activated with the two parameters previously defined.

Then, once everything is prepared, the server needs to be accepting connections and this is made with the listening function. After sending the listening command, the phone is able to be connected to the Raspberry Pi as a client.

Finally, once the phone is connected, it is compulsory to define the client and its address as a variable to be able to perform the communications later.

That is all the procedure that needs to be programmed in order to stablish the Raspberry Pi as a Bluetooth server. This process has been tested with the code in Testing codes. With this, then the user can send and receive data via Bluetooth using the following commands:

a. To send data:

```
client.send("data")
```

b. To receive data:

```
client.recv(1024)
#where the 1024 indicates the size in bits of the incoming text.
```

App to allow Bluetooth communication

Now, it has been presented how to connect the Raspberry Pi with the phone successfully but the phone also needs a software to stand this connection.

As it has been said many times, the design and programming of apps is not an objective of this project so, it will be used a simple software developed by the Massachusetts Institute of Technology to develop this app. This software is called App Inventor 2 and allows the user to create simple apps using a system of blocks of code [31] [32].

The app created has two main screens. The first screen only contains a brief summary of the instructions of use of the AiScope. Then the user has to click in the button proceed to go to the second screen.

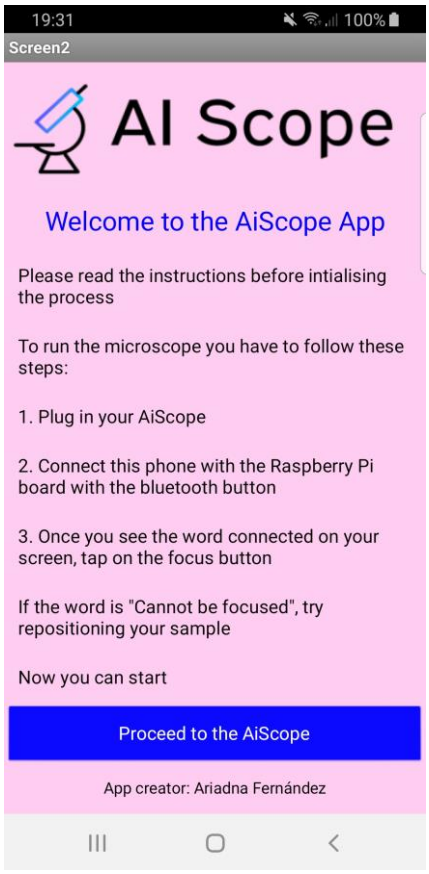


Figure 38 - Instructions screen of the AiScope App

This simple action only requires the following code block to be implemented:

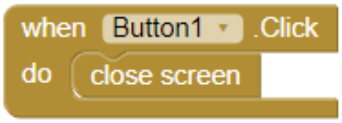


Figure 39 - Code block to close the screen when the button is clicked

The action described in this block is very descriptive, as it basically closes the screen when the user touches the button. Then, it opens the second screen.

The second screen has the main functionalities of the App. As it has been said, in the previous screen a short list of instructions is presented. It is important that these instructions are followed because the app commands need to be activated in the correct order.

Therefore, this screen is started to the following layout:

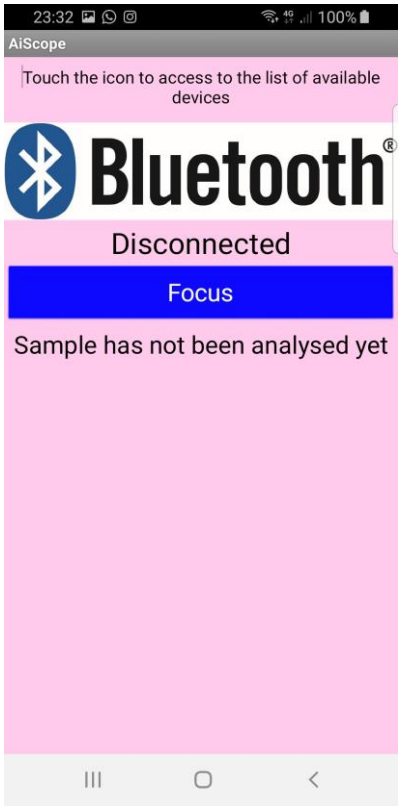


Figure 40 - Main screen of the App

When the screen is started, the first thing that it does is to initialise the global variable that will keep the orders received via Bluetooth. This way, this value is always present even though the app does not receive any data and prevents the following actions to crash the process.

This is coded in the following block:

Figure 41 - Initialisation of the variable to store the received data

Then, the first step to take in this screen is to connect the phone to the Raspberry Pi as a client. In order to do this, the app has a Bluetooth button that when pressed opens the list of paired devices:

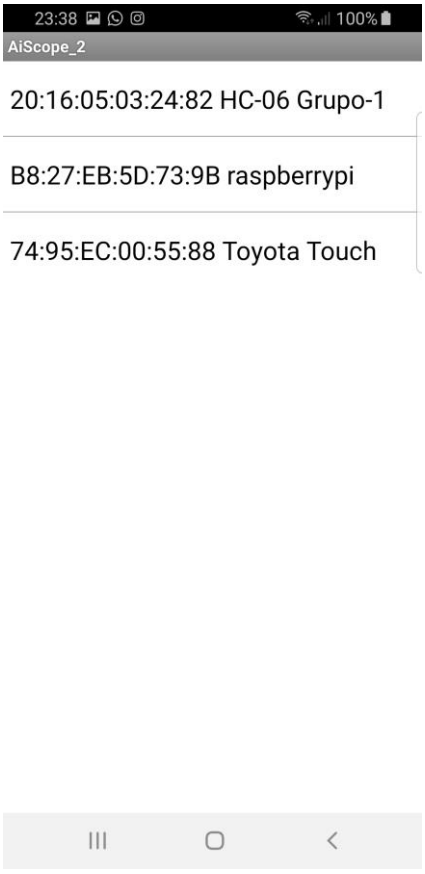


Figure 42 - List picker of the paired Bluetooth devices

There, the user only needs to select the Raspberry Pi. It is not difficult because every paired device includes its name and can be easily recognised. When this is selected, after a few seconds, the tag that previously indicated: Disconnected will change to one of these options:

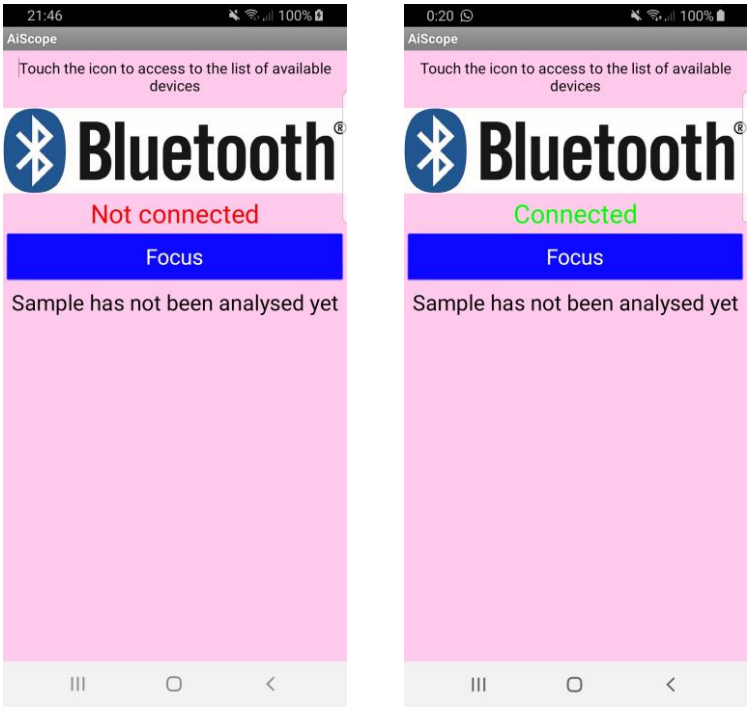


Figure 43 - When the user starts the App, a timer is fired. When the time is over, if the user has not started the Bluetooth connection, they will receive the first option. If the Bluetooth is connected, they will see the second option.

This process will take some seconds because a time margin is given in case it takes the user more than it is usual to select the connection. Then, when the label changes its status, the user has two options. If the label has changed to Connected, then the user can proceed and click the button focus. If not, and the label says Not connected, the user needs to try to stablish the connection again.

To complete this process, the App is designed with the following code blocks:

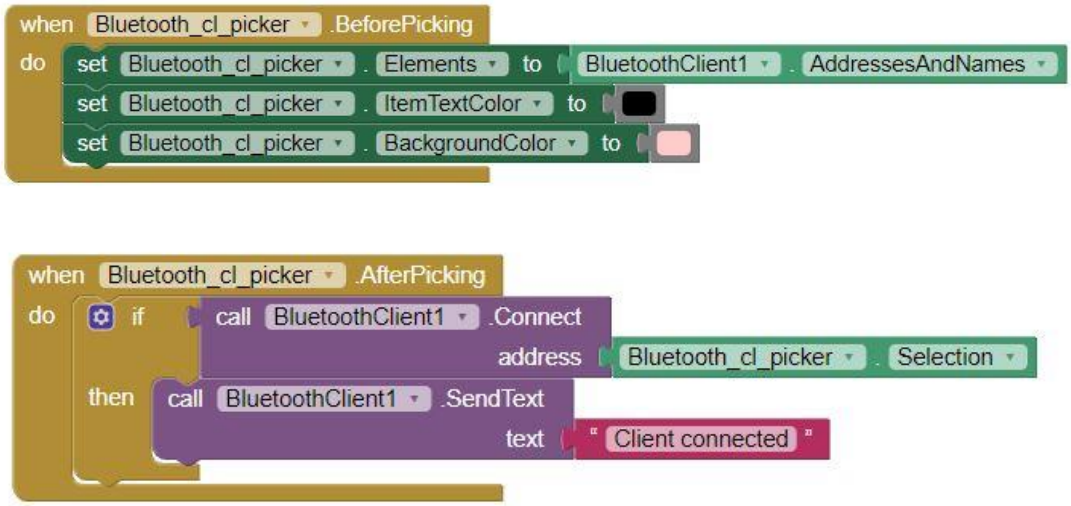


Figure 44 - Code blocks to connect the phone with Bluetooth

These blocks basically set the screen that is opened with the list of paired devices of the phone before it is opened, and after the user makes the selection, it sends a message to the server telling them that the client is connected. This message arrives to the Raspberry Pi with the function client.recv.

The last block of this part is the one that controls the connected/not connected label:

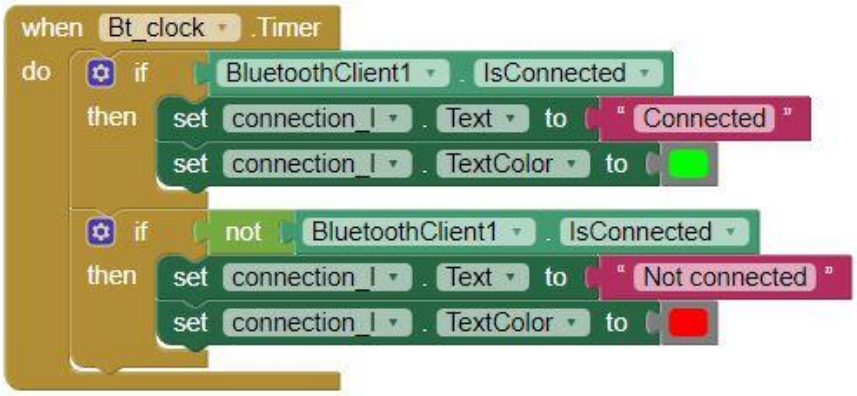


Figure 45 - Code blocks to change the label in function of the Bluetooth status

This works changing the value of the label to connected or not connected every time the counter gets to its time.

Then, and once the label is set to Connected, the user can click on the focus button. This will send a message to the Raspberry Pi telling them to start the focusing process and will set a new timer that will receive orders from the Raspberry Pi.

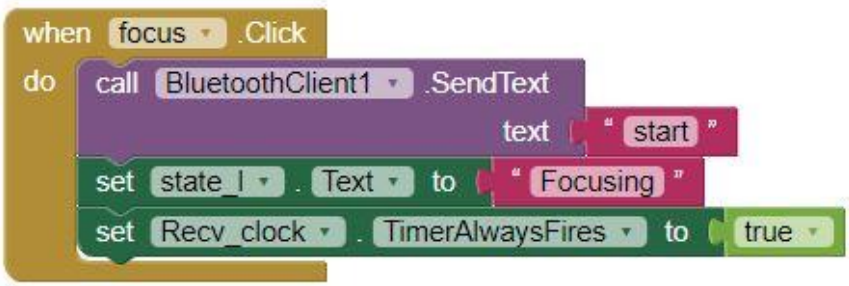


Figure 46- Sending a starting order to the Raspberry Pi when the user clicks on the Focus button

And this starts the main timer of the App. Then, the apps just receives data and executes orders. This can be seen in the following image:

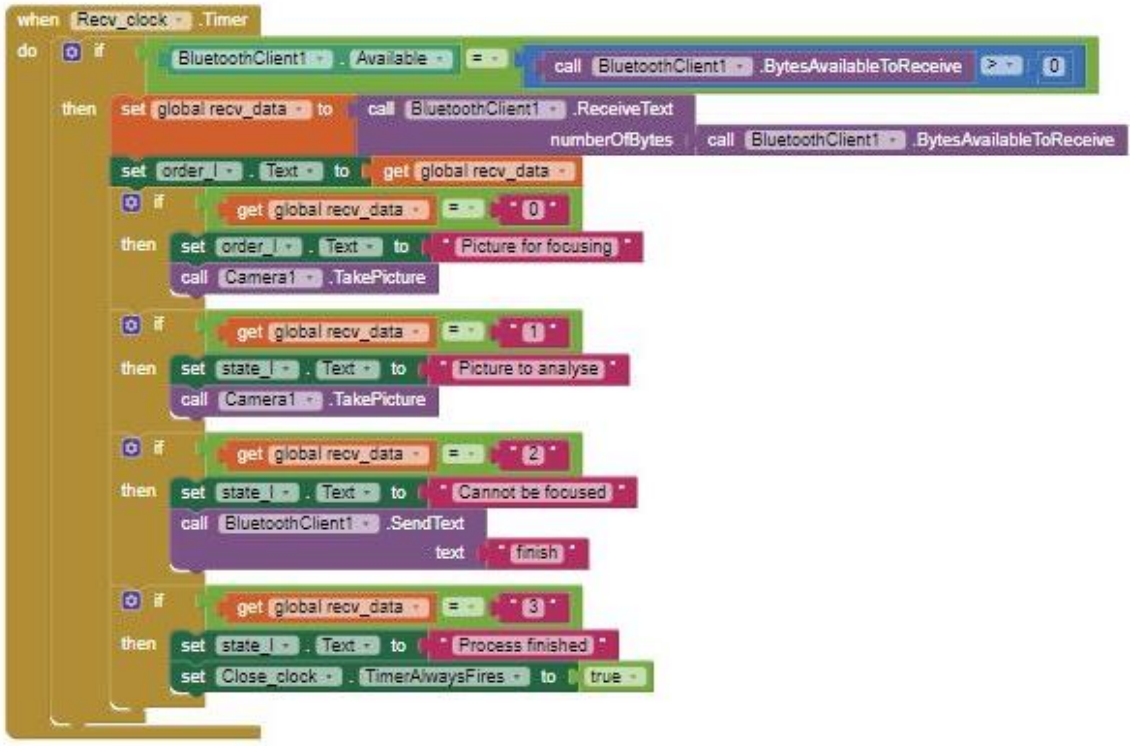


Figure 47 - Code block that analyses the orders received from the Raspberry Pi and does the required action

In this last picture, it can be seen how the App receives the orders from the Raspberry Pi. These orders are first stored as the global variable that was set when the scree was started and then are analysed by their content. Once they are analysed, depending on what they say, the App executes one action or another. In this case, the actions are restricted to take a picture with the camera if the order is 0, set the label to "Focused" and take an image too if the order is 1, set the label to "Cannot be focused" if the order is 2 and finish the process if the order is 3.

Here it must be pointed out that the user must click on the button of the camera every time a picture is taken. This is a huge inconvenience because it does imply that one person needs to be controlling the microscope the whole time but, at the same time, it does not affect to the fact that no trained personal will be needed to use the microscope.

Then, if the label is "Cannot be focused", this position of the sample will be discarded and the program will skip to the following one. When the process is finished the user will have to evaluate if the taken images are enough to evaluate that sample or if they need to start the process again. If the label indicates "Focused", the user is only informed that one useful picture has been taken and that in that moment, the sample will be repositioned and the focusing process will start again.

When the sample is focused, the user will be also shown the image taken. This is only made with an informative purpose but if the user knows how a focused sample has to look like, they can evaluate if the images are good and make adjustments to the reference values if not.

To show the image once the sample is focused, the following block is used:

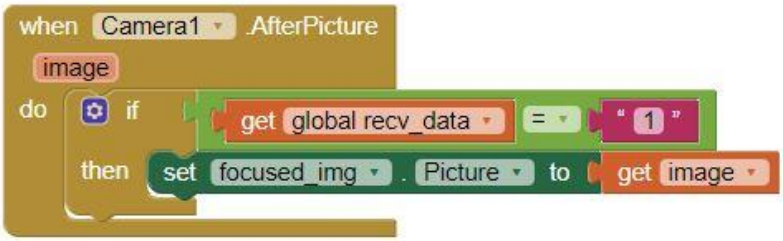


Figure 48 - Block that displays the captured image in the main screen

Finally, if the process is finished, the informative label is set to: "Process finished" and starts a timer. This timer is set to let the user a certain time to read the message and when the time is finished, it closes the app. The code block for that last action is the following one:

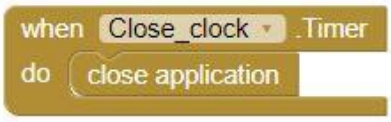


Figure 49 - Code block to close the App

File sharing via USB

The phone can be connected to the Raspberry Pi using its USB power cable. Once is connected and allowed to share multimedia files with the Raspberry Pi, it can access to any file in the phone.

The captured images are stored in the phone with a filename that does not seem recurrent. This hinders the access to the images. However, there are some possibilities to make the program able to recover them.

The selected option consists on indexing the files in the selected folder by the time they were added and then select the newest one. In order to do this, the following lines [33]:

```
import glob
import os


list_of_files = glob.iglob('Este equipo\Galaxy
S8\Phone\Pictures/*')
# * means all, if it was needed a specific format then it should
have been used *.csv
latest_file = max(list_of_files, key=os.path.getctime)
print (latest_file)
```

This is the preferable option; however, this can conduce to many errors in the code. In the experimental case, it works but as it has only been tested with one Raspberry Pi with a wide selection of complements installed, it cannot be ensured that it will work in any case.

However, the option of choosing the files manually does exist even though it requires the use of a screen and a keyboard or a VNC viewer App that allows to control the Raspberry Pi through the phone screen.

Final algorithm using the proposed solution and electronic assembly

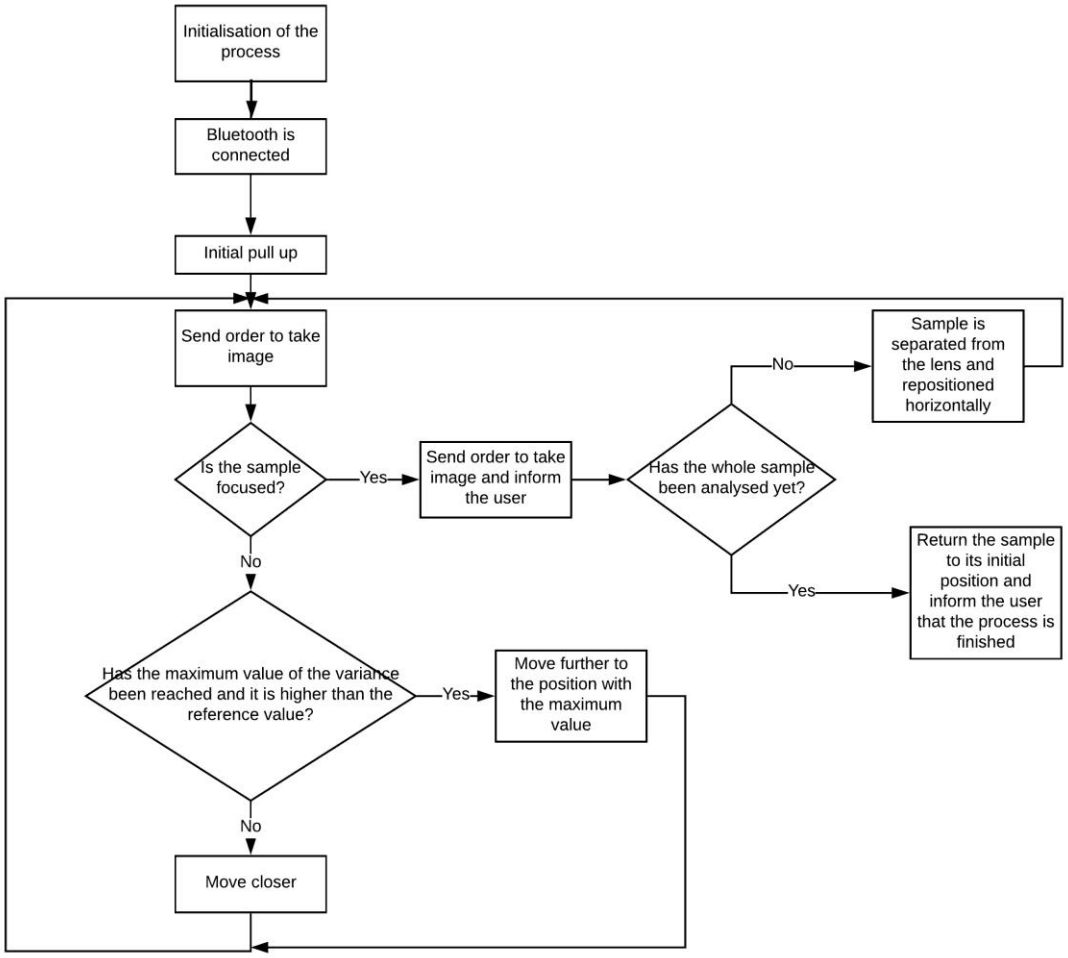


Figure 50 - Final algorithm flux diagram

It can be seen that the algorithm is not so different from the one presented in Figure 34; the only addition are the Bluetooth elements.

The same can be said about the electronic assembly built to make this work.

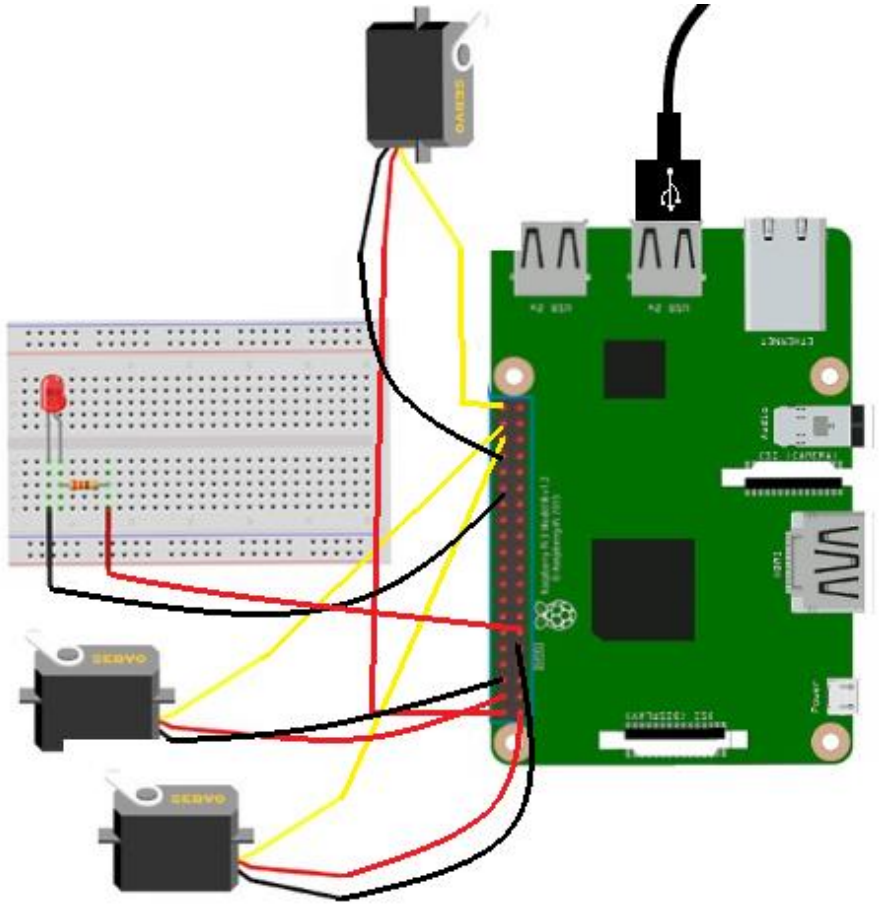


Figure 51 - Electronic assembly for the previously presented algorithm

It can be seen that the only changes made is the disappearance of the camera in the Raspberry Pi and the addition of the USB connector of the phone.

Likewise, it must be remarked the same that was previously said: even though here the led is connected using a protoboard, in the final prototype, the connection will not have it. Anyway, all the connections are exactly the same.

Reducing imprecisions in the working mechanism

One problem that arose with the displacement of the samples was a constant imprecision of the timings.

When the servomotors were connected to the Raspberry Pi and their respective GPIO channels were activated, they presented an erratic movement. This means, whilst they were supposed to be resting (with a duty cycle of 7.5%) they received sudden peaks of current that activated them for a moment or present a constant movement in a lower velocity. These movements that could not be neither predicted nor controlled, produced huge maladjustments in the timing of the

movements and were actually a problem while focusing the sample because once the process was finished, these movements could appear and blur the sample.

Therefore, to solve this problem, the first step was to analyse where it came from and then search which was the solution.

Analysis of the GPIO outputs

As it has been said, to analyse the problem, the first step was to see why the motor was acting that way. In order to see that, an oscilloscope was connected to the GPIO output that was controlling the servomotor to see if there was any problem.

Using the oscilloscope, it could be checked that the Raspberry Pi was providing the servomotor with the adequate duty cycles:

a. Duty cycle of 4.5%

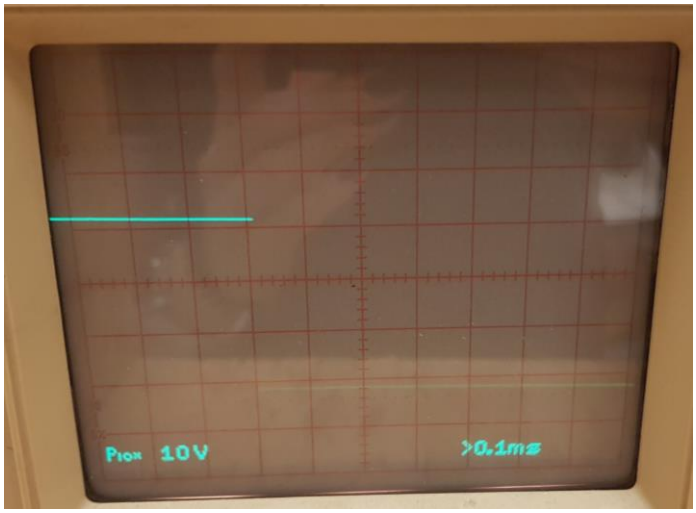


Figure 52- Observed signal when the output is a PWM signal with a Duty Cycle of 4.5%

b. Duty cycle of 7.5%

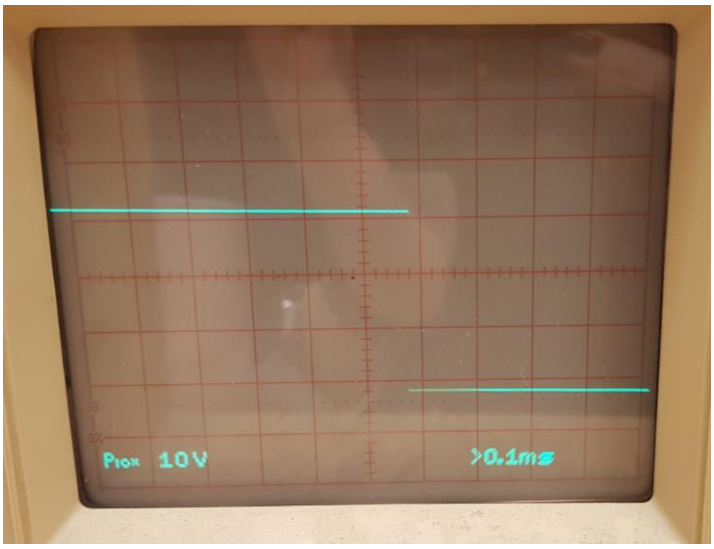


Figure 53- Observed signal when the output is a PWM signal with a Duty Cycle of 7.5%

c. Duty cycle of 10.5%

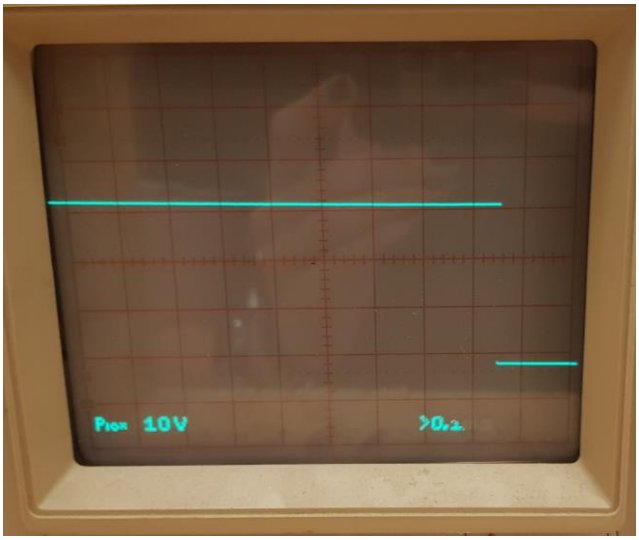


Figure 54 - Observed signal when the output is a PWM signal with a Duty Cycle of 10.5%

This did not give any clue of which the problem was so, different duty cycles were studied with a reduced time scale.

In a smaller time, scale, it could be observed that the cycles presented a tiny lag to the left every few cycles. This lag produced a small shake of the signal that was the thing producing the small movements in the servomotor.

This lag can be seen on the following images:

In this first figure, one can see how the PWM signal should be along the time:

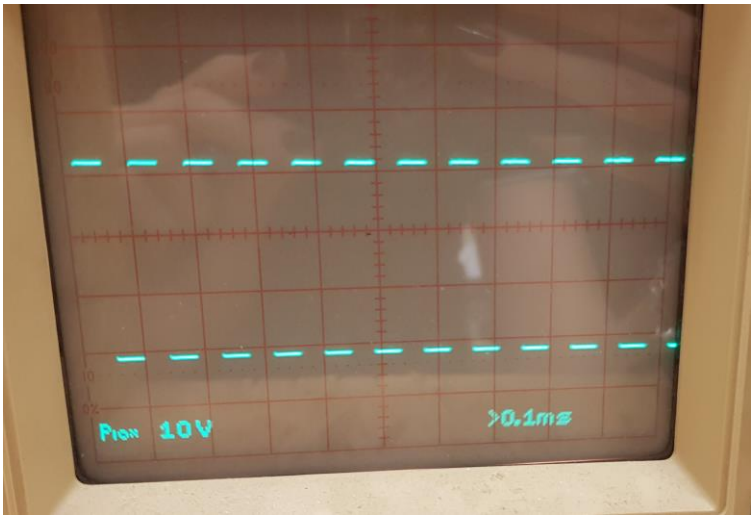


Figure 55 - Normal PWM signal

However, in determinate moments, the signal was distorted with a slight lag that caused the erratic movement of the servo:

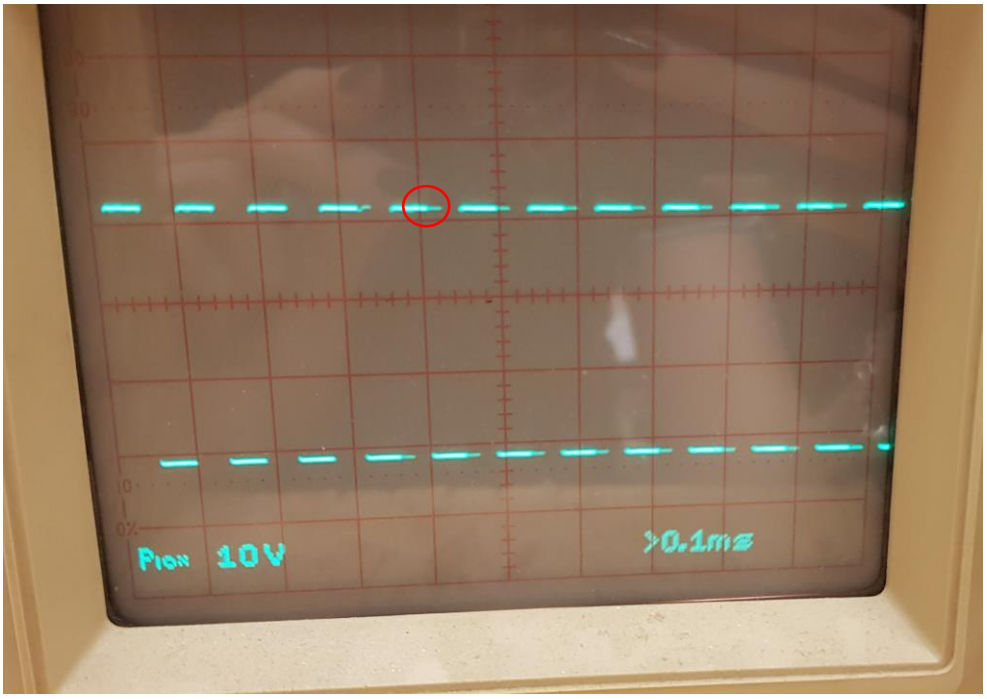


Figure 56- PWM signal with a small lag

This implied that the PWM cycle gave the demanded signal but in some cycles that signal was slightly delayed. That is why in the image one can see an enlargement of the signal.

This is due to the Raspberry Pi system interruptions. These interruptions are system checks that the program is running as it has to but can take up to 1ms. With a frequency of 50 Hz (which is actually really low), this can suppose until a 20% of the period.

Solutions for this problem

To solve this problem, two alternatives were proposed:

a. Forbid the interruptions in the Raspberry Pi system.
   This option would be the most accurate to solve this problem because basically prevents the Raspberry Pi from interrupting the code that is being executed. However, this requires modifying the basic structure and OS of the Raspberry Pi so it will not be considered.
b. Use the duty cycle 0 instead of the duty cycle 7.5 when the servomotors have to be resting.
   This solution is clearly less optimal than the previous one because involves more duty cycle changes and add several lines of code to the main program. Even though, it will be used because it does not need to modify the Raspberry Pi structure.

Therefore, this will be used to reduce the erratic movement of the servomotors.

This will work the following way:

When the servomotor was put to rest using the duty cycle 7.5, the slight displacement of the duty cycle could be seen, as it was shown in Figure 56. However, when the duty cycle is turned to 0, the servomotor is resting, but also does not receive any distortion on the signal because the signal is 0 for all the time:

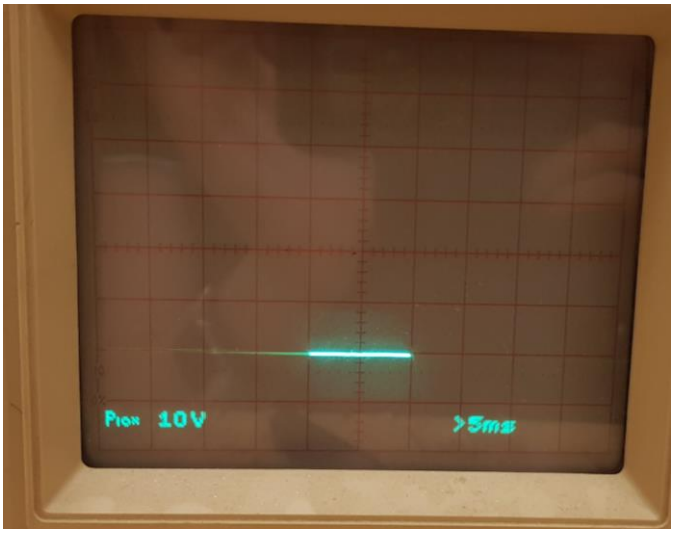


Figure 57 - Duty Cycle of a 0%

Chapter 5

Results

In this chapter, the results obtained with the development of the project will be presented. The project will be analysed economically and environmentally and will be presented a comparison between the initial and the final microscope. Finally, a future planning, to continue with the project, will be presented, with a series of improvements that could be made.

Economic summary

In this section, it will be presented a table to summarize the cost of building a single AiScope. For further information, on the project budget one can check the Budget document, which contains all the costs detailed. Here it will only detail the costs of the electronics that are the ones that have been added and treated in this project. Also, for further information one can check Joana Aina Martorell's project on the redesign of the AiScope [1].

So, at is has been said, in the following table one can see the whole cost of one AiScope:

Table 5 - Summary of the costs of one AiScope

	Units	**Cost/unit**	**Cost**
Materials			
Methacrylate plate (100x50 cm)	0.5	8.80€/plate	4.40€
DIN 975 M3 threaded rod	1	0.25€	0.25€
DIN 975 M5 threaded rod	1	0.33€	0.33€
Hexagonal nuts ISO 4036 d M3	5	0.02€	0.10€
Hexagonal nuts ISO 4036 d M5	32	0.03€	0.96€
Self-locking nuts DIN 985 d M3	5	2.25€/80 pack	0.14€
Self-locking nuts DIN 985 d M5	8	0.10€	0.80€
Silicone tube	20 cm	0.35€/m	0.07€
Flanges	3	6.99€/1000 pack	0.02€
Wing nuts DIN 315 d M5	2	3.75€/4 pack	1.88€
Polyethylene foam	0.5	1.9€/DIN A4 plate	0.95€
Nylon rounded bar	0.5	0.61€	0.31€
Ocular	1	13.19€	13.19€
Objective	1	19.20€	19.20€
Straight gear mod 0.5 D32Z	1	1.17€	1.17€
Straight gear mod 0.5 D46Z	2	1.17€	2.34€
Electronics			
Raspberry Pi Zero W	1	10€	10€
Micro SD card 32GB	1	5.99€/unit	5.99€
360º Servomotors	2	12.99€/5 pack	5.20€
180º servomotors	1	11.99€/5 pack	4.80€
Male-male connectors	2	5.83€/40 pack	0.30€
Male-female connectors	10	5.83€/40 pack	1.46€

	Units	Cost/unit	Cost
Female-female connectors	3	5.83€/40 pack	0.44€
Led	1	0.04€	0.04€
100Ω resistance	1	0.02€	0.02€
Total			**74.36€**

It is clear that the cost has been increased regarding to the initial one, which was around 51€, and this is significant to the development of the project. However, it is also a fact that electronic components need to be bought with a certain quality guarantee and that has a cost. Plus, it must be said that an addition of 20€ is not so significant if the prices of the AiScope and a real microscope are compared.

Environmental implications

This project will not have a huge environmental impact, as it does not contain any polluting element. The main source of energy that it requires is electricity, for both the assembly and the usage.

In order to avoid the generation of pollution with the existence of the AiScope, the ideal solution would be to only use electricity coming from renewable sources.

Apart from the energy, the other environmental implication of this project is the usage of materials. Between the different materials, some general categories with different environmental impacts can be distinguished:

a. Electronic components.
   Electronic components require a considerable amount of energy to be manufactured but also constitute an important source of waste because in the end of their useful life, they are normally abandoned in electronics cemeteries [34].
   At the level of this project, not so much can be done about the manufacturing of the electronic components, but yes after their useful life.
   Obviously, damaged components need to be replaced, but when an AiScope becomes useless, electronics in good state can be recycled, not only to make new AiScope microscopes but to any project that can require them. This way, the waste of resources is eliminated [35].
b. Methacrylate.
   The methacrylate is a recyclable plastic so once it has reached the end of its useful life, it can be send to a recycling plant where it is crushed, cleaned, homogenised and then, molten and moulded in new pieces. This way, again, the only impact that has the usage of this material is the required energy to produce it and then recycle it [36].
c. Steel.
   The steel is also a recyclable material that is actually widely recycled all along Europe. The recycle of the steel can help to reduce the amount of new resources required to produce new components and also help to reduce CO_2 emissions.
   It must be added, that it also requires energy to be manipulated and transformed into new pieces but the energy cost of recycling it is only a 30% of the energy cost of the production of new material, so recycling it, one is also saving energy [37].
d. Other small components.
   Elements like flanges or silicon are also used to build the AiScope.

Flanges are usually made of polyamides. These plastics are thermoplastics[8], which can be recycled. Therefore, even though they constitute a minimum part of the AiScope, they should be recycled.
Silicone is also a recyclable element, so the same procedure should be performed.
Polyurethane foam can also be recycled using chemical recycling, and even though it is not usual that it is transformed into the same product, it is used to produce isolating polyurethane foams that are widely used. This way, this does not constitute a residue.

Hence, it can be seen that almost all the elements that are recyclable and can have a second life after they are used. This way, there are only two things that need to be taken into account with this project. The first is to reduce and reutilise electronic components that are the most difficult ones to recycle. Moreover, the second is to optimise the manufacturing process, to require the minimum amount of energy and optimise the performance time of the microscope to also reduce the time it needs to be plugged in for a single analysis.

Conclusions and recommendations

After the work done, it is clear that the aim of the project has been achieved because the current AiScope prototype is automatic and can perform a whole sample analysis without the intervention of a trained person.

This prototype is an improved version of the initial one and it fulfils all the requirements. In the following table, there is presented a comparison of requirement achievement between the initial prototype and the final one.

Table 6 - Characteristics comparison

	Initial prototype	**Final prototype**
Automatic	No, the main requirement of this project was to make the microscope automatic	Yes, now the microscope can focus by itself and take the pictures. However, it does still need of human control through a mobile phone App.
Portable	Yes	Yes, its size has been a bit increased but it is still portable
Built in accessible materials	Yes	Yes. A few materials like silicone tubes or flanges have been added but these materials are easy to get anywhere. Regarding to the electronic part, all the materials should also be easy to get, as the Raspberry Pi is universal and the servomotors and led lights too.
Low-cost	Yes, one microscope costed about 51€	Yes, but a bit less. Now a microscope costs about 74€. The main increase of the price been the electronics but taking into account all the parts that have been added, the addition is not high.
Easy to manufacture	Yes, the main tool needed was a laser cutter but the pieces were designed for the	Yes, it is still needed the laser cutter but no additional building tools have been added. All the pieces are still designed

[8] Thermoplastics are materials that can be molten at high temperatures and become mouldable several times.

	Initial prototype	Final prototype
	possibility of not having one, so they could be sculpted.	to be able to be sculpted or cut and manufactured easily.
Easy to use	No, the user had to be an experimented technician that knew how to use a microscope.	Yes, anyone with a minimum knowledge on how to use a mobile phone can use it. Furthermore, using the manufacturing instruction written by Joana Aina Martorell [1] and the installing instructions found in Annex 3, anyone could also build it.

The requirements have been all achieved, as it can be seen in the table. However, also some improvements can still be made to get an even lower cost and a better portability and stability of the microscope. Moreover, in this case, it has not been possible to create a perfectly optimal software due to the lack of knowledge on how to program some things and deficiencies in the structure so it would be a pending task to improve the code.

In reference to the budget, it must be added that the cost of the project has been higher than it was expected because many options have been tried but at last, the final price is acceptable.

In conclusion, it can be said that the results of this project are good, because the main objective is done and now the microscope is automatic. Nevertheless, the microscope present some inefficiencies that should have been solved like the instability, the requirement of some human intervention in the process of taking the pictures or the simplicity of the App.

Future of the project

At this point, it has clearly been seen that this microscope is far from being a final model because it has many deficiencies and improvable aspects.

In the next iteration of the AiScope at least some points have to be compulsory treated.

In first place, the stability of the microscope has to be revised. Currently the stability of the AiScope is super fragile. The minimum movement or touch creates a real instability that makes the images taken to be blurry and, in some occasions even the sample to move or get unfocused.

Then, the horizontal movement mechanisms needs to revised too. It has to be redesigned thinking that the sample may need to be moved being in touch with the lens. Also the crank is connected to the piece that contains the sample in a poor way that is so imprecise and difficult to control.

Those points should be treated as a priority because solving them would reduce considerably the processing time and would actually also improve the quality of the taken images.

A part from that, there are some other points that could be treated. These ones are less important for the performance but are also advisable to consider to improve the quality of the AiScope and to reduce its price.

Another point that could be improved is the development of the App. In this case, as it was not a part of this project to learn App programming, an already existent App creator was used and this led to many limitations. It would be interesting to add some features like an option of manual focus and some option that allowed the user to see in real time what the camera was seeing and capture the desired areas.

Moreover, it would be ideal to develop an original App that could contain all the features that could help the user and improve the performance of the microscope. That way, that App could be

combined with the current App that is being developed by another AiScope collaborator who is programming a data base to store and classify the taken images.

Furthermore, and more related to hardware, another interesting point to treat would be to run the Python code in the phone with the App QPython. This would allow to change the Raspberry Pi from the model Zero W to just Zero because it would not require any Bluetooth communication. In this case, the Raspberry Pi would only control the GPIO outputs.

In all cases, as this is not going to be the last iteration of the microscope, many improvements can be performed in the following ones to get an even cheaper and better microscope. For that reason, a project Gantt for the next six months has been proposed:

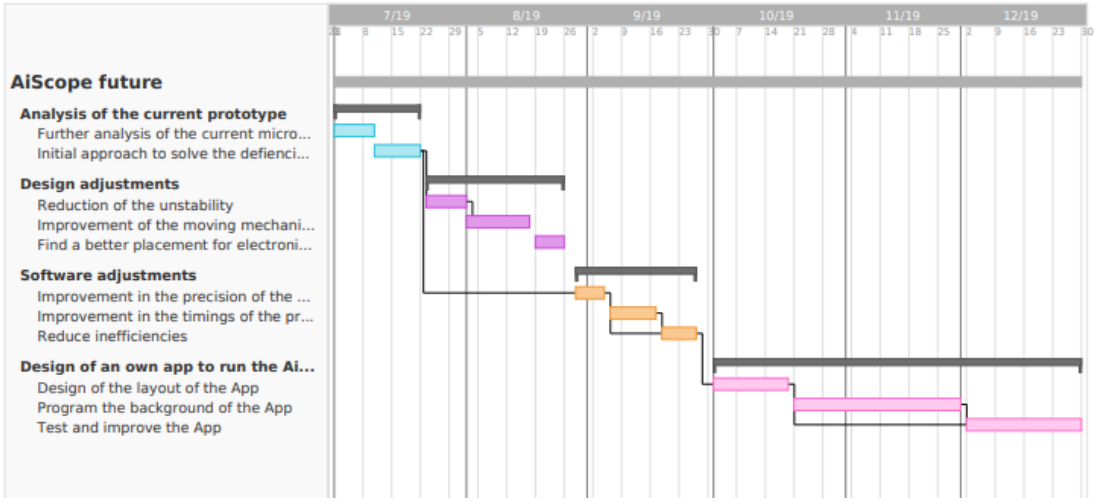


Figure 58 - Proposed Gantt diagram for the future of the project

It can be seen that there have been proposed all the improvements above mentioned. Those have been distributed in four main areas.

The first one includes a further analysis of the deficiencies in the current microscope. This will allow to explore the already detected deficiencies and to detect even more things to improve. If more things were considered to modify they could be added to this planning to be implemented.

Then, there are considered the design modifications. These ones should be made before the software is modified because and improvement in stability will directly have repercussions on the precision that can be obtained, so the order must be the considered one. Between the design modifications have been considered the stability, the moving mechanism and the electronics placement.

These are followed by software improvements that involve improving the precision of both the focusing system and the execution timings.

Finally, the last improvement that could be added is the creation of an original App to run the AiScope. This App should allow to run python codes and at the same time to take pictures without the need of a person clicking the camera button. It is noticeable that this part has a long period assigned because it is believed to be the longest one and also because App programming is a new thing introduced in this future iteration.

With this planning, a new prototype considerably improved could be available in the next months.

Bibliography

[1] J. A. Martorell, "Estudio y diseño de un microscopio de bajo coste para la detección de infecciones en muestras de sangre," Terrassa, 2019.

[2] "Ai Scope – Diagnosing global diseases with Computer vision," 2019. [Online]. Available: https://aiscope.net/. [Accessed: 26-Feb-2019].

[3] "Map of the Day: Where Malaria Has Been Defeated," *UN Dispatch*, 2017. [Online]. Available: https://www.undispatch.com/map-of-the-day-where-malaria-has-been-defeated/. [Accessed: 08-Jun-2019].

[4] "Diagnostic testing," *WHO*, 2018.

[5] "WHO | Microscopy," *WHO*, 2018.

[6] "Tinción de Giemsa - Blog de Laboratorio Clínico y Biomédico." [Online]. Available: https://www.franrzmn.com/tincion-de-giemsa/. [Accessed: 06-Mar-2019].

[7] Sheehan, "Tinción de Giemsa," *Blog de Laboratorio Clínico y Biomédico*, 1980. [Online]. Available: https://www.franrzmn.com/tincion-de-giemsa/. [Accessed: 06-Mar-2019].

[8] C. Wongsrichanalai, M. J. Barcus, S. Muth, A. Sutamihardja, and W. H. Wernsdorfer, "A Review of Malaria Diagnostic Tools: Microscopy and Rapid Diagnostic Test (RDT)," 2007.

[9] C.-C. for D. C. and Prevention, "CDC - Malaria - Diagnosis & Treatment (United States) - Diagnosis (U.S.)," 2019.

[10] "Improvement noted in malaria testing, diagnosis | Loop PNG." [Online]. Available: http://www.looppng.com/lifestyle/improvement-noted-malaria-testing-diagnosis-82872. [Accessed: 03-Mar-2019].

[11] "Microscopic Tests – Malaria Site." [Online]. Available: https://www.malariasite.com/microscopic-tests/. [Accessed: 07-Mar-2019].

[12] S. D. Pertuz Arroyo and H. R. Ibañez Grandas, "Ingeniería y Desarrollo.," *Rev. Científica Ing. y Desarro.*, vol. 22, no. 22, pp. 23–37, 2011.

[13] "Automated Microscopy," *Bitesize Bio*, 2017. [Online]. Available: https://bitesizebio.com/34349/automated-microscopy/. [Accessed: 31-May-2019].

[14] F. Paez, "Mini servo motor SG90 rotacion continua," *FPaez.com*, 2018. [Online]. Available: http://fpaez.com/mini-servo-motor-sg90-rotacion-continua/. [Accessed: 29-Apr-2019].

[15] L. Llamas, "Modelos de Raspberry Pi," *Luis Llamas*, 2017. [Online]. Available: www.luisllamas.es/modelos-de-raspberry-pi/. [Accessed: 16-May-2019].

[16] Adafruit, "Raspberry Pi B+, Pi 2, & Pi 3," 2019. [Online]. Available: https://www.adafruit.com/category/288. [Accessed: 24-Apr-2019].

[17] Adafruit, "Pi Zero v1.3 : Adafruit Industries," 2019. [Online]. Available: https://www.adafruit.com/category/934?src=raspberrypi. [Accessed: 24-Apr-2019].

[18] Ivmech, "Raspberry Pi Camera Module Multiplexer," *Amazon*, 2015. [Online]. Available: https://github.com/ivmech/ivport. [Accessed: 26-May-2019].

[19]! "Что такое Raspberry Pi - Raspberry Pi энтузиаст," *Electronicwings*, 2017. [Online]. Available: https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access. [Accessed: 18-May-2019].

[20] M. Á. Abellán, "Led intermitente - Control de GPIO con Python en Raspberry Pi," *Programo Ergo Sum*, 2018. [Online]. Available: https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/intermitente. [Accessed: 25-Apr-2019].

[21] "Puente H (L293D)," *Dinastía tecnológica*, 2019. [Online]. Available: http://dinastiatecnologica.com/producto/puente-h-l293d/. [Accessed: 18-May-2019].

[22] "Cómo controlar un motor con Raspberry Pi y L293D en Python," *Robologs*, 2017. [Online]. Available: https://robologs.net/2017/05/14/como-controlar-un-motor-con-raspberry-pi-y-l293d-en-python/. [Accessed: 03-Apr-2019].

[23] F. Paez, "Controlar un servomotor con Raspberry Pi," *FPaez.com*, 2018. [Online]. Available: http://fpaez.com/controlar-un-servomotor-con-raspberry-pi/. [Accessed: 31-Mar-2019].

[24] "Conversion de imagenes RGB a escala de grises con Python 3," *Python's eyes*, 2017. [Online]. Available: https://pythoneyes.wordpress.com/2017/05/22/conversion-de-imagenes-rgb-a-escala-de-grises/. [Accessed: 30-Mar-2019].

[25] C. Martínez and T. Martínez, "Detección de Bordes," pp. 1–32, 2012.

[26] C. Vision, "Image Gradients and Gradient Filtering 16-385 Computer Vision."

[27] A. Mordvintsev and K. Abid, "OpenCV-Python Tutorials Documentation," 2017.

[28] S. de la Fuente Fernández, "Modelos De Análisis De La Varianza," 2018.

[29] COMPAÑÍA LEVANTINA DE REDUCTORES, "Reductores de velocidad: aplicaciones y funcionamiento," *Blog CLR*, 2019. [Online]. Available: https://clr.es/blog/es/reductores-velocidad-funcionamiento/. [Accessed: 22-Apr-2019].

[30] Laurens-Wyuts, "Servo Motor Control With Raspberry Pi: 4 Steps," *instructables*, 2018. [Online]. Available: https://www.instructables.com/id/Servo-Motor-Control-With-Raspberry-Pi/. [Accessed: 26-May-2019].

[31] E. M, "Part 1: Basic Bluetooth communications using App Inventor," *App Inventor 2 - Learn to code*, 2015. [Online]. Available: https://appinventor.pevest.com/2015/01/23/part-1-basic-bluetooth-communications-using-app-inventor/. [Accessed: 25-May-2019].

[32] Laurens-Wyuts, "Control Arduino Using Android App: 8 Steps," *instructables*, 2017. [Online]. Available: https://www.instructables.com/id/Control-Arduino-using-android-app/#step5. [Accessed: 11-May-2019].

[33] "How to get the latest file in a folder using python," *Stack Overflow*, 2018. [Online]. Available: https://stackoverflow.com/questions/39327032/how-to-get-the-latest-file-in-a-folder-using-python. [Accessed: 09-Jun-2019].

[34] M. Martínez, "Basura electrónica, un problema del que muchos ciudadanos no saben deshacerse de manera correcta," *Nobbot*, 2018. [Online]. Available: https://www.nobbot.com/pantallas/donde-reciclar-basura-electronica/. [Accessed: 01-Jun-2019].

[35] D. Lorenzo Santana, "Cómo reciclar componentes electrónicos," *Tu taller de bricolaje*, 2013. [Online]. Available: https://tutallerdebricolaje.com/como-reciclar-componentes-electronicos/. [Accessed: 01-Jun-2019].

[36] A. Dumont-Fredo, "Comprar metacrilato en Madrid: ¿es posible reciclar metacrilato?," 2017. [Online]. Available: https://www.fredometacrilatomadrid.com/es/blog/se-puede-reciclar-el-metacrilato/. [Accessed: 01-Jun-2019].

[37] A. Fernández Muerza, "Para qué sirve reciclar acero y cómo hacerlo," *Ecodes*, 2010. [Online]. Available: http://www.consumer.es/web/es/medio_ambiente/urbano/2010/03/17/191793.php. [Accessed: 01-Jun-2019].

[38] Administración, "OpenCv y Raspberry Pi," *Booleanbite*, 2017. [Online]. Available: https://booleanbite.com/web/opencv-y-raspberry-pi/. [Accessed: 23-Apr-2019].

[39] "Python and Bluetooth," 2015. [Online]. Available: http://pages.iu.edu/~rwisman/c490/html/pythonandbluetooth.htm. [Accessed: 26-May-2019].

[40] "[TUTORIAL] Como ejecutar scripts al iniciar nuestra Raspberry," *Spainlabs.com*, 2016. [Online]. Available: https://www.spainlabs.com/foros/tema-TUTORIAL-Como-ejecutar-scripts-al-iniciar-nuestra-Raspberry. [Accessed: 28-Apr-2019].

Annex 1: Code used with the Raspberry Pi and the Raspberry Pi camera

Main program using the Raspberry Pi camera

```
#-----------------------------------------------------------------
#-AISCOPE WORKING CODE ---------------------------------------
#-Author: Ariadna Fernández Martínez ----------------------------
#-----------------------------------------------------------------

#import the libraries
from gpiozero import LED
from time import sleep
import RPi.GPIO as gpio

#set up of the pin mode (this will determine the naming of the pins)
gpio.setmode(gpio.BCM)

#first step will be to turn on the led that is required to see the sample
#intialsation of the led
led=LED(17)

#led will be on throughout all the process
led.on()

#set up of the pins that will be used as the servomotor output
gpio.setup(16,gpio.OUT)
gpio.setup(20,gpio.OUT)
gpio.setup(21,gpio.OUT)
#set up of the PWM frequency: 50 Hz
z = gpio.PWM(16,50)
x = gpio.PWM(20,50)
y = gpio.PWM(21,50)
#initial position of the servomotor
z.start(0)
```

```
x.start(0)
y.start(7.5)

#initialisation of the time counters (in seconds)
t=20
tlim=22

#initialisation of the position counter
p=2.5

#the sample starts in the position 0, which is the furthest from
#the objective so the user can change the sample properly without
#damaging the objective
#Therefore, the first part is to move it up close to the objective
z.ChangeDutyCycle(10.5)
sleep(t)
z.ChangeDutyCycle(0)
#this will move the sample up for t seconds
#and then the loop of positions is started

while p<12.5:

    #first the position of the crank is set
    y.ChangeDutyCycle(p)

    #Focus the sample
    tfoc=process(t,tlim)
    tmove=tfoc-t
    z.ChangeDutyCycle(4.5)
    sleep(tmove)
    z.ChangeDutyCycle(0)
    t=tfoc

    #Take a picture
    take_pic()

    #Then the sample is slightly separed from the lens to move it
    z.ChangeDutyCycle(4.5)
    sleep(2)
```

```
        z.ChangeDutyCycle(0)

        #and the rod is half-turned
        x.ChangeDutyCycle(7.75)
        sleep(0.58)
        x.ChangeDutyCycle(0)

        #the process is started again
        tfoc=process(t,tlim)
        tmove=tfoc-t
        z.ChangeDutyCycle(4.5)
        sleep(tmove)
        z.ChangeDutyCycle(0)
        t=tfoc

        #Take a picture
        take_pic()

        #Then the sample is slightly separed from the lens to move it
        z.ChangeDutyCycle(4.5)
        sleep(2)
        z.ChangeDutyCycle(0)

        #and the rod is half-turned
        x.ChangeDutyCycle(7.75)
        sleep(0.58)
        x.ChangeDutyCycle(0)

        p=p+0.5

#once it has finished, the sample is returned to the position 0
z.ChangeDutyCycle(4.5)
sleep(t)
z.ChangeDutyCycle(7.5)

#gpio channel is cleared to prevent from imprecisions
gpio.cleanup()
# and the led is turned off
led.off()
```

Functions used in the previous program

Process of focusing the sample

```
#focusing process function

def process(t,tlim):

    #initialisation of the lists that will keep the values
    time=[]
    modvar=[]

    #then is taken the first value for the focus evaluation
    md=focus()
    modvar += md
    time += t

    while t<tlim:
        z.ChangeDutyCycle(10.5)
        sleep(0.125)
        z.ChangeDutyCycle(0)
        md=focus()
        modvar += md
        t=t+0.125
        time += t

    #that loop will move the sample up for tlim-t periods depending
    #on the distance range that one wanted to study

    #once the values are obtained the maximum is searched
    maxvar=max(modvar)
    #then its position on the list and the time it was reached
    posmax=modvar.index(maxvar)
    tfoc=time(posmax)

    return(tfoc)
```

Taking pictures of the sample block and analysing them

```
#image analysis function

def focus():

    #package initialisation
    import cv2
    from picamera import PiCamera
    import numpy as np
    import math

    #initialisation of the camera
    camera=PiCamera()

    #taking a picture
    camera.capture('image_name.jpg')

    #analysis of the image

    #1. importing the captured image
    img=cv2.imread('imagename.jpg',0)

    #2. calculation of the sobel derivatives
    sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=3)
    sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=3)
    sobel=np.abs(sobelx)+np.abs(sobely)

    #3. calculation of the variance of the sobel derivatives
    var1=np.var(sobel)

    #4. calculation of the modified variance of the sobel derivatives
    mod1=(var1**2)*(n/(n-1))

    #5. returning the value
    return math.sqrt(mod1)
```

Taking one picture

```
#taking a picture function

def take_pic():

    #package initialisation
    from picamera import PiCamera

    #intialisation of the camera
    camera=PiCamera()

    #taking a picture
    camera.capture('defimage_number.jpg')
```

Testing codes

Moving up and down the sample (test)

```
#functions declaration
import RPi.GPIO as gpio
from time import sleep
from gpiozero import LED

#set up of the pin mode (this will determine the naming of the pins)
gpio.setmode(gpio.BCM)
#set up of the pin that will be used as the servomotor output
gpio.setup(16,gpio.OUT)
#set up of the PWM frequency: 50 Hz
p = gpio.PWM(16,50)
#initial position of the servomotor
p.start(7.5)

#set up for the led
led=LED(17)
#turning on the led
```

```
led.on()

#initialisation of the time counters (in seconds)
t=20
tlim=23

#the sample starts in the position 0, which is the furthest from
#the objective so the user can change the sample properly without
#damaging the objective
#Therefore, the first part is to move it up close to the objective
p.ChangeDutyCycle(10.5)
sleep(t)
p.ChangeDutyCycle(7.5)
#this will move the sample up for t seconds

while t<tlim:
    p.ChangeDutyCycle(10.5)
    sleep(0.125)
    p.ChangeDutyCycle(7.5)
    t=t+0.125

#that loop will move the sample up for tlim-t periods depending on the
#distance range that one wanted to study

#finally, the sample is returned to the position 0
p.ChangeDutyCycle(4.5)
sleep(tlim)
p.ChangeDutyCycle(7.5)

#gpio channel is clear to prevent from imprecisions
gpio.cleanup()
```

Analysis of a sequence of images (test)

```
#functions declaration
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
import pandas as pd
import math

#Here it is calculated the number of pixels of the images
#This is different depending on the camera
n=1080*1912

for i in range(number_of_images):
    c=i+1
    img=cv2.imread('nuevas/img%d.jpg' %c,0)
    sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=3)
    sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=3)
    sobel=np.abs(sobelx)+np.abs(sobely)
    var1=np.var(sobel)
    mod1=(var1**2)*(n/(n-1))
    print(math.sqrt(mod1))

#this will print the modified variance of each image of the
sequence
#Then these values can be directly exported to excel to be analysed
```

Moving the sample horizontally (test)

```
#first step will be to import the libraries needed to execute the
#program
from time import sleep
import RPi.GPIO as gpio

#set up of the pin mode (this will determine the naming of the
pins)
gpio.setmode(gpio.BCM)

#set up of the pins that will be used as the servomotor output
gpio.setup(20,gpio.OUT)
gpio.setup(21,gpio.OUT)
#set up of the PWM frequency: 50 Hz
x = gpio.PWM(20,50)
y = gpio.PWM(21,50)
```

```
#initial position of the servomotor
#position x (the rod) is initialised in 0 (resitng)
x.start(0)
#position y (the crank) is initialised on the right
y.start(2.5)

#Then the sequence of movements is initialised

p=2.5
while p<12.5:
    #first the crank is moved
    y.ChangeDutyCycle(p)
    #here the first picture would be taken
    #then the rod is half-turned to get the other position
    x.ChangeDutyCycle(7.75)
    sleep(0.58)
    x.ChangeDutyCycle(0)
    #here the second picture would be taken
    #and then the rod is returned to the other side
    x.ChangeDutyCycle(7.75)
    sleep(0.58)
    x.ChangeDutyCycle(0)
    p=p+0.5

#To finish the script the GPIO is closed
gpio.cleanup()
#And would make a total of 40 pictures of the sample
```

Annex 2: Code used with the Raspberry Pi and the mobile phone camera

Main program using the mobile phone camera

```
#-----------------------------------------------------------------
#-AISCOPE WORKING CODE V2----------------------------------------
#-Author: Ariadna Fernández Martínez ----------------------------
#-----------------------------------------------------------------

#the code works in two main parts, one to connect the Raspberry
#Pi with the phone via bluetooth and the second one that makes the
#whole process of focusing and moving the sample

#first step will be to import the libraries needed to execute the
#program
from tkinter import Tk
from tkinter.filedialog import askopenfilename
from time import sleep
import RPi.GPIO as gpio
import bluetooth
import os
import glob
import cv2
import numpy as np
import math
from gpiozero import LED

#set up of the pin mode (this will determine the naming of the
pins)
gpio.setmode(gpio.BCM)

#initialisation of the led
led=LED(17)

#the led will be on throughout all the process
```

```
led.on()

#-------------------------------------------------------------
#first part: connecting the Raspberry via bluetooth
#-------------------------------------------------------------

#First it need to be stablished the name of the port
host = ""
port = 1    # Raspberry Pi uses port 1 for Bluetooth Communication

# Creaitng Socket Bluetooth RFCOMM communication
server = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
#The Raspberry Pi will be ther server and the phone the client

server.bind((host,port))

server.listen(1) # One connection at a time

# Server accepts the clients request and assigns a mac address.
client,address = server.accept()

#This checks that the data is arriving
Con_data=client.recv(1024)

#-------------------------------------------------------------
#second part: focusing the sample and taking pictures
#-------------------------------------------------------------

#set up of the pins that will be used as the servomotor output
gpio.setup(16,gpio.OUT)
gpio.setup(20,gpio.OUT)
gpio.setup(21,gpio.OUT)
#set up of the PWM frequency: 50 Hz
z = gpio.PWM(16,50)
y = gpio.PWM(20,50)
x = gpio.PWM(21,50)
#initial position of the servomotor
z.start(0)
x.start(0)
```

```
y.start(2.5)

#initialisation of the time counters (in seconds)
t=16
tlim=19

#initialisation of the position counter
p=2.5
pp=0

#the sample starts in the position 0, which is the furthest from
#the objective so the user can change the sample properly without
#damaging the objective
#Therefore, the first part is to move it up close to the objective
z.ChangeDutyCycle(10.5)
sleep(t)
z.ChangeDutyCycle(0)
#this will move the sample up for t seconds

#then is taken the first value for the focus evaluation

#first the order of taking a picture is sent
client.send("0")
#then the path of the picture is obtained
path="\Galaxy S8\Phone\Pictures\"
paths=".jpg"
filename=get_latest_file(path,paths)

var=0
var1=focus2(filename)

status="not focused"

while p<12.5:

    #first the position of the crank is set
    y.ChangeDutyCycle(p)

    while pp<2:
```

```
#Then the position of the rod is set
x.ChangeDutyCycle(7.75)
sleep(0.58)
x.ChangeDutyCycle(0)

#and the distance to the lens is reduced
z.ChangeDutyCycle(10.5)
sleep(1)
z.ChangeDutyCycle(0)
t=t+1

while t<tlim and status=="not focused":

    if var1>370:
        client.send("1")
        sleep(60)
        var1=0
        var=0
        status="focused"

    elif var1<370 and var1>=var:

        z.ChangeDutyCycle(10.5)
        sleep(0.125)
        z.ChangeDutyCycle(0)

        client.send("0")
        sleep(60)
        var=var1
        filename=get_latest_file(path,paths)
        var1=focus2(filename)
        t=t+0.125

    elif var1<370 and var1<var:

        z.ChangeDutyCycle(4.5)
        sleep(0.0625)
        z.ChangeDutyCycle(0)
```

```
                client.send("0")
                sleep(60)
                var=var1
                filename=get_latest_file(path,paths)
                var1=focus2(filename)
                t=t-0.0625


        if t>=tlim and status=="notfocused":
            client.send("2")
            sleep(60)
            break

        status="not focused"
        pp=pp+1

        #the sample is moved down to be repositioned
        z.ChangeDutyCycle(4.5)
        sleep(2)
        z.ChangeDutyCycle(0)
        t=t-2

    pp=0
    p=p+0.5

client.send("3")
sleep(60)

#finally the sample is returned to the position 0
z.ChangeDutyCycle(4.5)
sleep(t)
z.ChangeDutyCycle(0)

# and the led is turned off and the servomotors closed
led.off()
gpio.cleanup()
```

Functions for the previous program

Function for image analysis

```
#image analysis function

def focus2(filename):

    #analysis of the image

    #1. importing the captured image
    img=cv2.imread("\GalaxyS8\Phone\Pictures\%s.jpg"
    %filename,0)

    #2. calculation of the sobel derivatives
    sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=3)
    sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=3)
    sobel=np.abs(sobelx)+np.abs(sobely)

    #3. calculation of the variance of the sobel
    var1=np.var(sobel)

    #4. calculation of the modified variance of the sobel
    mod1=(var1**2)*(n/(n-1))

    #5. returning the value
    return math.sqrt(mod1)
```

Function to find the last file of a directory

```
#finding last file of a directory
def get_latest_file(path, *paths):
    """Returns the name of the latest (most recent) file
    of the joined path(s)"""
    fullpath = os.path.join(path, *paths)
    list_of_files = glob.iglob(fullpath)
```

```
    if not list_of_files:
        return None
    latest_file = max(list_of_files, key=os.path.getctime)
    _, filename = os.path.split(latest_file)
    return filename
```

Testing codes

Bluetooth connection and data transfer (test)

```
# Importing the Bluetooth Socket library
import bluetooth

host = ""
port = 1
# Raspberry Pi uses port 1 for Bluetooth Communication

# Creaitng Socket Bluetooth RFCOMM communication
server = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
print('Bluetooth Socket Created')

server.bind((host, port))
print("Bluetooth Binding Completed")

server.listen(1) # One connection at a time

# Server accepts the clients request and assigns a mac address.
client,address = server.accept()
print("Connected To", address)
print("Client:", client)

# Receivng the data.
data = client.recv(1024) # 1024 is the buffer size.
print(data)
#Sending the data
send_data=”0”
client.send(send_data)
```

Annex 3: Installing instructions for the Raspberry Pi

All the information in the following section can be found on the internet but is has been considered to include it in an Annex of this project because the future AiScope manufacturer will need to do all these procedures before the microscope can be used. Therefore, it will useful for them to have all the information recompiled in this section so they do not forget to add anything.

Installing Noobs (Raspberry Pi OS) will not be included as it only requires to be downloaded and copied into the micro SD card (official Raspberry Pi micro SD cards already come with it). Then the installation process is guided and intuitive.

Here it will be presented how to install the required libraries to run the codes here presented, how to configure the Bluetooth connection and how to make the code to be executed when the Raspberry Pi is plugged in.

Libraries installation

OpenCV

To install OpenCV in Raspberry Pi the following code from the blog Boolean bite [38] has to be followed using the command window:

1. The first step is to update the system:

   ```
   $sudo apt-get update
   $sudo apt-get upgrade
   ```

2. Then it must be installed the dependencies to compile the libraries

   ```
   $sudo apt-get install build-essential cmake pkg-config
   ```

3. A series of libraries required.

   ```
   $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
   $ sudo apt-get install libxvidcore-dev libx264-dev
   $ sudo apt-get install libgtk2.0-dev
   $ sudo apt-get install libatlas-base-dev gfortran
   ```

4. And the Python interface

   ```
   $ sudo apt-get install python3-dev
   ```

5. Once all the libraries are installed, one need to download the code of OpenCV and its library opencv_contrib

```
$ wget -O opencv.zip
https://github.com/opencv/opencv/archive/3.2.0.zip

$ unzip opencv.zip

$ wget -O opencv_contrib.zip
https://github.com/Itseez/opencv_contrib/archive/3.2.0.zip

$ unzip opencv_contrib.zip
```

6. Now it is required to create a virtual environment to use the libraries when they are needed. In order to do this, it is necessary to install pip, Python's package controller; to create the environment and finally prepare it to install opencv.

```
$ wget https://bootstrap.pypa.io/get-pip.py

$ sudo python get-pip.py

$ sudo pip install virtualenv virtualenvwrapper

$ sudo rm -rf ~/.cache/pip

$ echo -e "\n# virtualenv and virtualenvwrapper" &gt;&gt;
~/.profile

$ echo "export WORKON_HOME=$HOME/.virtualenvs" &gt;&gt;
~/.profile

$ echo "source /usr/local/bin/virtualenvwrapper.sh" &gt;&gt;
~/.profile

$ source .profile

$ mkvirtualenv cv -p python3

$ workon cv

(cv) user@localdomain$_

$ pip install numpy

$ cd ~/opencv-3.2.0/

$ mkdir build

$ cd build

$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
    -D BUILD_EXAMPLES=ON ..
```

7. Once it is ready, it needs to be compiled and installed

```
$ make -j4


$ make install
$ ldconfig
```

8. Finally it is necessary to change its name to make it accessible

```
$ sudo mv cv2.cpython-34m.so cv2.so
```

```
cd ~/.virtualenvs/cv/lib/python3.4/site-packages/
$ ln -s /usr/local/lib/python3.4/site-packages/cv2.so cv2.so
```

9. It can be checked if the installation is correct with the following Python lines. If it returns the version of cv2 installed, all is correct.

```
import cv2
cv2.__version__
```

Individual libraries

Required libraries to make the AiScope work are the following ones:

a. Bluetooth
b. Numpy (included in OpenCV)
c. Cv2 (included in OpenCV)
d. Math
e. GPIO
f. Time
g. Os
h. Glob

To install these libraries, if they were missing one only needs to introduce the following command in the command window with the Raspberry Pi connected to Internet. Pip installer should be already able because it was installed with OpenCV.

```
sudo pip install name of the library
```

Bluetooth pairing

Bluetooth pairing and connection must be checked before running the code for the first time. In order to do it, the user will have to run the following lines in the command window [39]:

```
hciconfig
#this just in case it was not installed from before
sudo apt-get install python-dev libbluetooth-dev
sudo service Bluetooth status
#locate Bluetooth service and then edit
Sudo nano /lib/system/system/Bluetooth.service
#insert –C after bluetoothd in this script, then save and exit
Sudo reboot
#restart the system to apply the changes
#one can also check if the changes have been applied again with
#sudo service Bluetooth status
```

```
Sudo bluetoothctl
#the following commands are used inside bluetoothctl
Power on
Pairable on
Discoverable on
Agent on
Default-agent
Trust #name of device
#then exit the bluetoothctl environment and continue
Wget https:/bootstrap.pypa.io/get-pip.py
#in case pip was not installed
Sudo python get-pip.py
Git clone https://github.com/karulis/pybluez.git
#obtain pybluez for Python
Cd pybluez/
Sudo python setup.py install
Sudo nano Bluetooth_adv
#in this file add the following lines
Sudo hciconfig hci0 piscan
Sudo sdptool add SP
#then save and exit
#continue on the command window
Sudo chmod +x Bluetooth_adv
Sudo ./bluetooth_adv
Sudo python pybluez/examples/simple/rfcomm-server.py
```

After that, the Raspberry Pi should not present any problem to connect with any Bluetooth device.

Start the code when the Raspberry Pi is started

Running the code when the Raspberry Pi is started will allow the user to avoid using VNC viewer with their phone or having to user a keyboard and a screen to control the Raspberry Pi. This has not been implemented in the current prototype as it is not necessary to avoid the use of any material and it is more useful, for testing, to see the code and why it is not working. However, this is proposed to be used in the microscopes that are sent to isolated places to simplify the utilisation.

In order to do this, once everything is set correctly, the following steps must be followed with the AiScope code script [40]:

1. Execute the following command:

```
sudo nano /etc/init.d/detector-init
```

This will create a new document on the route /etc/init.d/ where the following code must be copied:

```
#! /bin/sh
# /etc/init.d/detector-init

### BEGIN INIT INFO
# Provides:          detector-init
# Required-Start:    $all
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: example script of automatic initialising
# Description:
### END INIT INFO

# Depending on the parameters the user must use one or other option
case "$1" in
 start)
   echo "Running detector-init"
   # Here the user inserts the name of their script
   /usr/bin/python /home/pi/detector.py
   ;;
 stop)
   echo "Deteniendo detector-init"

   ;;
 *)
   echo "Use mode: /etc/init.d/detector-init {start|stop}"
   exit 1
   ;;
esac

exit 0
```

2. Then this file is given permission to be executed:

```
sudo chmod 755 /etc/init.d/detector-init
```

3. Check that everything works properly:

```
sudo /etc/init.d/detector-init start
```

4. And finally, the automatic initialisation is activated:

```
sudo update-rc.d detector-init defaults
```

Annex 4: Experimental data

In this section will be presented the tests performed to establish a reference value about the focus of the image. The tests were performed in basis of six series of images taken while a pilot program to focus the image was being executed. This program, as it has been said, was a pilot program, that is the reason why there are series of pictures that start before others (the motors usually were stuck while moving up and down and that made them to be uncoordinated) or why they have more images than others do. However, these series are representative of how the microscope will work, because it does need to be prepared at any point (it must be taken into account that in the future, people can move it and cause these imprecisions.

Therefore, the values of the modified variance for the Laplacian derivative and the Sobel derivative with which the Table 2 and Table 3 have been created are presented in the following tables.

Table 7 - Values of the modified variance of the Sobel derivatives applied to a series of pictures of a sample

	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6
1			244,95203		136,04047	
2			222,04275		242,89630	
3			212,81022		290,46202	
4			187,20818		270,61737	
5			209,78114		287,30325	
6			192,17066	335,34700	245,36151	
7	360,58093	479,67369	249,81994	308,99760	263,23992	
8	310,50603	467,31574	185,01200	350,88404	286,33198	223,23194
9	336,22616	421,60591	231,93581	352,86599	252,04677	317,11634
10	371,37520	488,96853	189,15908	340,58669	281,45121	250,07214
11	326,43373	425,24061	202,25939	290,51571	255,34957	244,33538
12	319,49007	524,18724	256,47346	284,70552	224,32406	252,82442
13	333,40474	554,47279	198,51031	308,22959		471,86665
14	447,61548	599,66323	224,25358	288,07748		657,58808
15	396,00030	600,61530	539,07083	554,06250	267,42137	675,60178
16	629,03679	685,41692	582,21962	588,76480	270,43049	625,84714
17	608,50813	717,09261	560,50722	583,46398	314,04997	740,40994
18	649,75676	736,31733	661,32303	586,46674	608,47993	695,28307
19	825,33487	781,55052	704,76634	655,86914	692,75880	540,84546
20	874,57776	830,49387	728,06483	777,85471	767,15100	599,55081
21	261,42878	499,67737	357,58770	604,25601	268,78379	488,68773
22	249,98249	498,37144	341,38187	609,57514	333,89334	413,48862
23				498,27620	304,57417	463,93437
24				433,22165		415,23496
25				223,91724		583,82189

Table 8 - Values of the modified variance of the Laplacian applied to a series of pictures of a sample

	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6
1			11,1204815		10,3335810	
2			16,2918796		15,7041689	
3			14,5230369		9,7634650	
4			11,7855898		13,2171485	
5			10,3416254		11,9005266	
6			7,8972609	13,2717036	15,7653792	
7	28,3627376	29,0150774	6,6070885	8,8283167	14,2834079	
8	13,4774682	13,1079209	10,8321565	13,0194786	16,5083659	6,1469887
9	10,4273585	10,6487494	9,9438266	13,4564590	10,3847424	6,8879971
10	10,1107140	12,9338373	11,2191807	7,5325269	8,3109118	10,6506065
11	9,1371107	10,9896849	11,4546236	11,1300062	7,9301283	9,5060107
12	9,2865573	16,2176396	9,5856639	9,4050331	7,0765577	8,4146025
13	11,0815907	16,5410010	9,0470837	13,7083460		16,0488111
14	15,5863737	18,4057329	7,9124785	10,4242100		17,1240596
15	12,5223413	17,4978899	16,3609018	16,6033225	9,7213053	18,6614379
16	18,1259939	17,8561691	17,8656890	14,8811893	11,2736643	14,9586740
17	19,4956654	17,2228939	13,9689394	14,8309670	13,1291665	15,6233404
18	15,9463760	17,3003283	16,3032267	16,3163255	13,7515305	15,0136145
19	15,5328844	16,4014533	18,3297055	16,3566445	16,8026452	15,2651779
20	17,4072138	16,7035268	19,5461860	16,6738503	16,1212423	15,9880527
21	10,1988592	11,6374100	15,4321419	11,8744693	11,9678985	14,6882367
22	10,6748061	15,2014820	10,1967180	16,1002856	13,9760719	16,3039181
23				12,2691881	13,9591834	14,3009500
24				11,7837115		14,9859534
25				10,2925235		15,7381763

Notice that the tables have some values marked in colours. These colours indicate:

a. Red: point where the sample starts showing
b. Orange: point where the red cells starts showing
c. Green: focused sample
d. Blue: over-focused sample

Notice that this pictures were taken with a stable support instead of the designed support for the phone (see reference [1]) for more precision to see which method was valid. However, those were not going to be the real values obtained when real analysis were performed because neither the support was going to be that stable nor the microscope itself is stable due to the constant motion of the servos and the general vibration they generate.

Then, to solve this, it was also performed a test that consisted on testing different images now taken with the phone in the microscope with different reference values in order to fix one to determine whether the image was focused or not.

In this case, 148 images, both focused and not focused were tested. The results are presented in the following table:

Table 9 - Values of the modified variance used to obtain a reference value

Number of image	Focused	Value of the modified variance of the gradient
1	No (too close)	278
2	No (too close)	276
3	No (too far)	182
4	No (too far)	201
5	No (too far)	176
6	No (too far)	196
7	No (too far)	175
8	Yes	402
9	Yes	407
10	Yes	405
11	No (too far)	308
12	No (too far)	317
13	No (too far)	198
14	No (too far)	204
15	Yes	365
16	Yes (blurry)	264
17	Yes (blurry)	309
18	Yes (blurry)	323
19	Yes (blurry)	327
20	Yes	400
21	Yes	437
22	No (too far)	185
23	No (too close)	191
24	No (too far)	207
25	No (too close)	220
26	No (too far)	174
27	No (too far)	181
28	No (too close)	183
29	No (too far)	245
30	No (too far)	240
31	No (too close)	205
32	No (too far)	146
33	No (too far)	121
34	Yes (blurry)	158
35	No (too far)	139
36	No (too far)	136
37	No (too far)	185
38	No (too far)	187
39	No (too far)	193
40	No (too far)	192
41	No (too far)	188
42	Yes (blurry)	266
43	No (too close)	236
44	No (too close)	206
45	No (too close)	239
46	No (too far)	220
47	No (too far)	220
48	No (too far)	215

Number of image	Focused	Value of the modified variance of the gradient
49	Yes (blurry)	266
50	Yes	311
51	No (too far)	195
52	No (too far)	209
53	No (too far)	216
54	No (too far)	228
55	No (too far)	208
56	No (too close)	223
57	Yes	282
58	Yes	274
59	Yes	286
60	Yes	319
61	No (too close)	258
62	No (too close)	260
63	Yes	317
64	No (too far)	278
65	Yes	316
66	Yes	379
67	No (too far)	256
68	No (too far)	245
69	No (too far)	238
70	No (too far)	233
71	No (too far)	249
72	No (too far)	237
73	No (too far)	243
74	No (too far)	227
75	No (too close)	243
76	Yes (blurry)	262
77	Yes	320
78	Yes	300
79	No (too close)	352
80	No (too close)	334
81	No (too close)	332
82	No (too close)	337
83	No (too close)	337
84	No (too close)	344
85	No (too close)	338
86	No (too close)	326
87	No (too close)	342
88	No (too far)	288
89	No (too far)	280
90	No (too far)	270
91	No (too far)	277
92	No (too far)	281
93	No (too far)	288
94	No (too far)	285
95	No (too far)	240
96	No (too far)	225
97	No (too far)	234
98	No (too close)	244
99	No (too close)	244

Number of image	Focused	Value of the modified variance of the gradient
100	No (too close)	243
101	No (too close)	246
102	No (too close)	240
103	No (too far)	251
104	No (too close)	260
105	No (too close)	260
106	No (too close)	267
107	No (too far)	212
108	No (too far)	206
109	No (too far)	208
110	No (too close)	217
111	No (too far)	209
112	No (too far)	256
113	No (too far)	255
114	No (too far)	240
115	No (too far)	227
116	No (too far)	264
117	No (too far)	277
118	No (too far)	257
119	Yes	317
120	Yes	316
121	Yes	347
122	No (too far)	225
123	No (too far)	236
124	No (too far)	216
125	No (too far)	232
126	No (too far)	223
127	No (too close)	254
128	No (too far)	222
129	No (too far)	233
130	No (too close)	339
131	No (too close)	348
132	No (too close)	368
133	Yes	466
134	Yes	482
135	Yes	482
136	No (too far)	295
137	No (too far)	298
138	No (too far)	268
139	No (too far)	262
140	No (too far)	295
141	No (too far)	299
142	Yes	436
143	Yes	488
144	Yes	453
145	No (almost focused)	317
146	No (almost focused)	367
147	No (too far)	285
148	No (too far)	260